

INORGANIC THIN FILMS AND NANOSHEETS: FABRICATION,  
CHARACTERIZATION AND SIMULATION

A Dissertation

Presented to the Faculty of the Graduate School  
of Cornell University

In Partial Fulfillment of the Requirements for the Degree of  
Doctor of Philosophy

by

Mahmut Aksit

August 2014

© 2014 Mahmut Aksit

# INORGANIC THIN FILMS AND NANOSHEETS: FABRICATION, CHARACTERIZATION AND SIMULATION

Mahmut Aksit, Ph. D.

Cornell University 2014

Two-dimensional (2D) inorganic materials present exciting opportunities for new scientific and technological breakthroughs. In this work, novel fabrication characterization and simulation techniques are presented for inorganic nanosheets and nanostructured thin films with the motivation of advancements in thermoelectrics, flexible electronics, optoelectronics and thermal engineering.

Metal oxide nanosheet stacks of  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  are fabricated using a novel bottom-up method based on sol-gel chemistry, E-field induced kinetic demixing and high temperature heat treatment. The nanosheet thicknesses can be 10–100 nm while their lengths can measure up to 1.8 mm long. The stacked nanosheets are readily delaminated into very large ( $<350 \mu\text{m}$  long,  $\sim 100$  nm thick) free-standing 2D crystals. Both  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets are electrically conductive and show ductility. Thermoelectric efficiency of bulk  $\text{Na}_x\text{CoO}_2$  is expected to improve in the nanosheet form due to phonon confinement and scattering.

Novel p-type TCO thin films of  $\text{Ca}_3\text{Co}_4\text{O}_9$  nano-plates are produced using a sol-gel and spin coating based process. The process parameters can be varied to produce TCO thin films with sheet resistance as low as  $5.7 \text{ k}\Omega/\text{sq}$  ( $\rho \approx 57 \text{ m}\Omega \cdot \text{cm}$ ) or with average visible range transparency as high as 67%. The FOM for the top-performing  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin film ( $151 \text{ M}\Omega^{-1}$ ) is higher than FOM values reported in the literature for all other solution processed, p-type TCO thin films and higher than most

others prepared by PVD and CVD.

Frequency resolved phonon transport experiments are performed on nanofabricated Si nanosheets using micro-scale phonon spectrometry devices. Current work mainly focus on understanding the frequency resolved phonon transport measurement results using Monte Carlo (MC) simulations. These MC simulations assume that phonon transmission is dominated by phonon-surface interactions and use the well-known Ziman theory to predict phonon-surface scattering rates. Although, the MC model predicts a diffuse surface scattering probability of less than 40% for the measured surface roughness (1 nm), the measurements are consistent with a 100% probability. The nanosheets therefore exhibit the so-called ‘Casimir limit’ at a much lower frequency than expected if the phonon scattering rates follow the Ziman theory.



## BIOGRAPHICAL SKETCH

Mahmut Aksit was born on October 17, 1984 in Ankara, the capital of Turkey and he grew up in a smaller city called Denizli in Southwest Anatolia. He attended schools in this small city through high school. Mahmut, then, moved to Istanbul for his undergraduate education at Sabanci University. When he first started his undergraduate education, Mahmut's intention was to become a mechanical engineer like his father and his brother in law. He later recognized that he likes science more than engineering and hoping that he would be more involved in science he took lots of courses related to micro-electronics. As he learned more about the operation of micro-electronic devices, Mahmut started to think that revolutionary advancements in micro-electronics technology are strongly tied to advancements in materials chemistry and physics. Therefore he selected Material Science and Engineering as a major and Physics as his minor. During his undergraduate education, he had the chance to preform research with brilliant scientists who migrated to Turkey from the USA and Soviet Russia. This way, he was able to see that there can be lot of different approaches to scientific research. He also learned that what he can do with materials science knowledge is not limited to micro-electronics technology.

After completing his undergraduate education in Sabanci University, he started his Ph. D. program under the supervision of Professor Richard D. Robinson in the Department of Materials Science and Engineering at Cornell University in Ithaca, New York. Under Professor Robinson's supervision, he had chance to work on many different topics related to Materials Chemistry and Applied Physics.

To my wife and parents for their love and support

## ACKNOWLEDGMENTS

I would first like to express my deepest appreciation to my advisor, Professor Richard D. Robinson. This work would not be possible without his excellent guidance, scientific knowledge and tremendous support. His attention to details, leadership and hard work have always set an example to me as a researcher in science and engineering. His consistent encouragement has been very important for me to overcome many frustrations throughout this research. Prof. Robinson's optimism and enthusiasm for new ideas has helped my research to evolve in many different directions in a highly interdisciplinary and enjoyable manner.

I would also like to thank Professor Frank DiSalvo and Professor Bruce van Dover for serving on my thesis committee, their valuable advice and for providing the various scientific equipment that made my research possible.

I am very grateful to my parents and family for fueling my interest in science at an early age and supporting me throughout. Thanks also to all my friends for supporting me on the bad days, and joining me to celebrate the good ones.

Finally, I thank Ayse Nur, my loving and supportive wife for, making me smile even on the bad days, helping to keep me sane and being there for me all along.

## TABLE OF CONTENTS

BIOGRAPHICAL SKETCH.....	v
DEDICATION .....	vi
ACKNOWLEDGMENTS.....	vii
TABLE OF CONTENTS .....	viii
LIST OF FIGURES.....	xi
LIST OF TABLES .....	xix
LIST OF ABBREVIATIONS .....	xx
1. INTRODUCTION.....	1
1.0. Nanosheets and Nanostructured Thin Films for Practical Applications .....	1
1.1. Metal Oxide Nanosheets.....	6
1.1.0. Synthesis.....	6
1.1.0.0. Chemical Exfoliation.....	6
1.1.0.1. Vapor-Liquid-Solid Method.....	7
1.1.0.2. Bottom-up Solution Growth Methods.....	9
1.1.1. Device Integration .....	10
1.2. p-type Transparent Conducting Oxide Thin films.....	12
1.2.0. Solution-based Synthesis Methods.....	13
1.2.0.0. Spray Pyrolysis.....	13
1.2.0.1. Spin Coating .....	13
1.3. Nanoscale Phonon Transport.....	14
1.3.0. Phonon Spectrometry Experimental Setup.....	16
1.3.1. Modeling of Phonon Spectrometry Through Nanosheets .....	20
REFERENCES .....	21
2. SCALABLE NANOMANUFACTURING OF MILLIMETER-LENGTH 2D	
Na <sub>x</sub> CoO <sub>2</sub> NANOSHEETS .....	27
2.0. Abstract.....	27
2.1. Introduction .....	28
2.2. Experimental.....	30
2.2.0. Synthesis of Na <sub>0.7</sub> CoO <sub>2</sub> nanosheets .....	30
2.2.1. Exfoliation of the nanosheets .....	31
2.2.2. XRD, SEM, EDS and TEM characterization .....	32
2.2.3. Synchrotron x-ray analysis .....	32

2.3. Results and Discussion .....	33
2.4. Conclusions .....	52
REFERENCES .....	53
3. SYNTHESIS AND PROPERTIES OF ELECTRICALLY CONDUCTIVE, DUCTILE, EXTREMELY LONG (~50 $\mu\text{m}$ ) NANOSHEETS OF $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$ .....	57
3.0. Abstract.....	57
3.1. Introduction .....	57
3.2. Experimental.....	61
3.2.0. Synthesis of $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$ nanosheets.....	61
3.2.1. Delamination of the nanosheets.....	63
3.2.2. XRD, SEM, EDS, WDS and TEM characterization .....	63
3.2.3. Electrical Resistivity Measurements .....	64
3.3. Results and Discussion .....	65
3.4. Conclusion.....	86
REFERENCES .....	87
4. MISFIT LAYERED $\text{Ca}_3\text{Co}_4\text{O}_9$ AS A HIGH FIGURE OF MERIT p-TYPE TRANSPARENT CONDUCTING OXIDE FILM THROUGH SOLUTION PROCESSING .....	92
4.0. Abstract.....	92
4.1. Introduction .....	92
4.2. Methods .....	94
4.2.0. Synthesis of p-type TCO Thin Films.....	94
4.2.1. Measurement and Evaluation of Transparent Conductivity .....	96
4.3. Results and Discussion .....	97
4.4. Conclusion.....	107
REFERENCES .....	108
5. DIRECT MEASUREMENTS OF SURFACE SCATTERING IN Si NANOSHEETS USING A MICROSCALE PHONON SPECTROMETER: IMPLICATIONS FOR CASIMIR-LIMIT PREDICTED BY ZIMAN THEORY ....	111
5.0. Abstract: .....	111
5.1. Introduction .....	112
5.2. Results and Discussion .....	119
5.3. Conclusion.....	147
REFERENCES .....	148
6. CONCLUSIONS AND OUTLOOK .....	155

6.0. Millimeter Length Metal Oxide Nanosheets Synthesis.....	155
6.1. P-type TCO Thin Films of Misfit layered $\text{Ca}_3\text{Co}_4\text{O}_9$ .....	160
6.2. Micro-scale Phonon Spectrometry through Si Nanosheets .....	163
REFERENCES .....	168
A. SUPPLEMENTARY INFORMATION FOR CHAPTER 2 .....	169
B. SUPPLEMENTARY INFORMATION FOR CHAPTER 3 .....	180
B.0. Supplementary Methods: Area Detector XRD Sample Mounting .....	191
REFERENCE(S) .....	191
C. SUPPLEMENTARY INFORMATION FOR CHAPTER 4 .....	192
D. SUPPLEMENTARY INFORMATION FOR CHAPTER 5 .....	197
D.0. Supplementary Discussion: Casimir-Ziman Surface Scattering Regimes .....	197
D.1. Supplementary Discussion : Detailed Explanation of Phonon Pathways.....	198
D.2. Supplementary Discussion: Description of Measurement of Background Level <i>nbs</i> .....	200
D.3. Supplementary Discussion: Additional Details of Monte Carlo Calculations of Phonon Focus Behavior.....	204
D.4. Supplementary Methods: Generation of 2D phonon focusing factor map for angles relative to (110) direction, using the Monte Carlo Approach .....	206
D.5. Supplementary Discussion: Comparison of Monte Carlo and Analytic Calculations of Line-of-Sight Ballistic Phonon Transmission Factor.....	208
D.6. Supplementary Methods: Monte Carlo C++ Code .....	219
D.7. Supplementary Methods: Analytical Model Matlab Code .....	290
REFERENCES .....	301

## LIST OF FIGURES

<b>Figure 1.1: AFM images and height profiles of chemically exfoliated <math>\text{Ti}_{0.91}\text{O}_2</math>, <math>\text{MnO}_2</math> and <math>\text{Ca}_2\text{Nb}_3\text{O}_{10}</math> nanosheets. ....</b>	<b>7</b>
<b>Figure 1.2: SEM images of <math>\text{Ga}_2\text{O}_3</math> nanosheets and nanobelts produced by VLS growth. ....</b>	<b>8</b>
<b>Figure 1.3: SEM images of <math>\text{NiCo}_2\text{O}_4</math> precursor nanosheet network. ....</b>	<b>9</b>
<b>Figure 1.4: Thickness dependent maximum dielectric constants of various promising dielectric materials compared with dielectric constants of <math>\text{Ti}_{0.87}\text{O}_2</math> nanosheets. ....</b>	<b>11</b>
<b>Figure 1.5: FET characteristics of titania nanosheet layer-by-layer assembly. ...</b>	<b>11</b>
<b>Figure 1.6: Representative drawing of STJ based microscale phonon spectrometry device. ....</b>	<b>15</b>
<b>Figure 1.7: a) and b) Band diagram of quasiparticle tunneling and phonon decay processes in source and detector STJs. c) Diagram showing phonon emission from emitter, propagation through transport medium and detection at the detector. ....</b>	<b>18</b>
<b>Figure 1.8: Representative diagram for phonon emission. ....</b>	<b>19</b>
<b>Figure 1.9: Schematic diagram of phonon transport measurement. ....</b>	<b>19</b>

<b>Figure 2.1: SEM and optical microscope images of metal oxide nanosheets:.....</b>	<b>34</b>
<b>Figure 2.2: Schematic diagram for the synthesis procedure for millimeter-length 2D Na<sub>x</sub>CoO<sub>2</sub> nanosheets:.....</b>	<b>37</b>
<b>Figure 2.3: (a) SEM images of the flakes that appear after the autocombustion of the PAA-Metal Complex aqueous solution. The thickness of the flakes is &lt; 200 nm and the length measures up to 400 μm. (b) SEM images of the edge-to-edge- connected Na<sub>x</sub>CoO<sub>2</sub> crystals that appear after the low temperature calcination (650 °C) of the autocombusted form.....</b>	<b>39</b>
<b>Figure 2.4: The effect of kinetic-demixing. ....</b>	<b>43</b>
<b>Figure 2.5: Na-deficient and Na-rich regions after E-field induced kinetic- demixing performed under nitrogen gas (a) and atmospheric conditions (b). ....</b>	<b>46</b>
<b>Figure 2.6: X-ray Characterization of Nanosheets.....</b>	<b>47</b>
<b>Figure 2.7: TEM images of nanosheets of Na<sub>0.7</sub>CoO<sub>2</sub>.....</b>	<b>49</b>
<b>Figure 3.1: TEM, HR-TEM and optical microscopy images of delaminated K<sub>x</sub>CoO<sub>2</sub>·yH<sub>2</sub>O nanosheets. ....</b>	<b>67</b>
<b>Figure 3.2: Temperature dependent electrical resistivity values for K<sub>x</sub>CoO<sub>2</sub>·yH<sub>2</sub>O nanosheets (blue dots). ....</b>	<b>70</b>
<b>Figure 3.3: Schematic diagram of the nano-manufacturing procedure for millimeter-length K<sub>x</sub>CoO<sub>2</sub>·yH<sub>2</sub>O nanosheets. ....</b>	<b>71</b>



<b>Figure 3.4: X-ray powder diffraction for each step of the nanosheet synthesis procedure (ordered from top to bottom).....</b>	<b>73</b>
<b>Figure 3.5: Characterization of the processing by electric-field induced kinetic-demixing. ....</b>	<b>75</b>
<b>Figure 3.6: SEM and optical images of stacked nanosheets. ....</b>	<b>79</b>
<b>Figure 3.7: XRD of a single stack of water intercalated nanosheets.....</b>	<b>82</b>
<b>Figure 3.8: XRD analysis of <math>K_xCoO_2 \cdot yH_2O</math> nanosheet stack(s) using area detector:.....</b>	<b>85</b>
<b>Figure 4.1: TCO thin film of p-type <math>Ca_3Co_4O_9</math> obtained by spin coating and calcination of viscous resin on quartz substrate. ....</b>	<b>99</b>
<b>Figure 4.2: Sheet resistance, specular optical transmission and FOM values for <math>Ca_3Co_4O_9</math> TCO thin films obtained with different solute concentrations and calcination times. ....</b>	<b>102</b>
<b>Figure 4.3: XRD and XPS of TCO thin films. ....</b>	<b>106</b>
<b>Figure 5.1: Design of Spectrometer to Probe Phonon Transmission through Silicon Nanosheet Arrays.....</b>	<b>117</b>
<b>Figure 5.2: Phonon Spectrometer Operating Principles.....</b>	<b>120</b>
<b>Figure 5.3: Phonon Transmission Spectra through Nanosheet Arrays.....</b>	<b>128</b>

<b>Figure 5.4: 2D Phonon Focusing Factor Map of Angles Relative to Si [110] Direction Generated by MC Algorithm. ....</b>	<b>131</b>
<b>Figure 5.5: Measured Phonon Transmission Signals through Un-etched (bulk) Si Mesa and Comparison to Simulated Transmission Factors.....</b>	<b>133</b>
<b>Figure 5.6: Measured Phonon Transmission Signals through Si Nanosheet Arrays: Comparison to Modeled Transmission Factors. ....</b>	<b>140</b>
<b>Figure 5.7: Estimated Transmission Through Si Nanosheet, as a Function of Nanosheet Length, for Phonon Distribution Peaked at 400 GHz. ....</b>	<b>143</b>
<b>Figure A.1: Additional SEM images for nanosheet metal-oxides. ....</b>	<b>169</b>
<b>Figure A.2: Histogram of thickness of 100 layers as measured by SEM.....</b>	<b>170</b>
<b>Figure A.3: X-ray powder diffraction of the <math>\text{Na}_x\text{CoO}_2</math> between each step of the synthesis procedure. ....</b>	<b>171</b>
<b>Figure A.4: Schematic diagram representing anisotropic growth of nano-layers at different calcination temperatures without kinetic demixing (a) and with kinetic demixing (b).....</b>	<b>172</b>
<b>Figure A.5: (a) Schematic representation of the Grazing Incidence Diffraction setup in G line at CHESS.....</b>	<b>174</b>
<b>Figure A.6: Schematic of the real (a) and reciprocal (b) space representation of</b>	

the $\text{Na}_x\text{CoO}_2$ lattice.....	175
<b>Figure A.7: Optical microscope image taken in reflected (a) and transmitted (b) white light of a large exfoliated nanosheets.....</b>	<b>176</b>
<b>Figure A.8: TEM images showing single layers on the edge of an exfoliated nanosheets. ....</b>	<b>177</b>
<b>Figure A.9: Preliminary thermoelectric measurements (electrical conductivity, Seebeck coefficient and thermal conductivity) of <math>\text{Na}_{0.7}\text{CoO}_2</math> (blue data points) and <math>\text{Na}_{0.9}\text{CoO}_2</math> (red data points) nanosheet stacks. ....</b>	<b>178</b>
<b>Figure B.1: Top view optical image of the configuration for the two-point resistivity measurements of <math>\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}</math> nanosheet stacks. ....</b>	<b>180</b>
<b>Figure B.2: SEM images and EDS of the exfoliated <math>\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}</math> nanosheet in Figure 3.1c (main text) .....</b>	<b>181</b>
<b>Figure B.3: (a) HR-TEM images of two different exfoliated <math>\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}</math> nanosheets at different magnifications. (b) Moire fringes in (a) revealed using Edge Finder function of ImageJ software .....</b>	<b>182</b>
<b>Figure B.4: Electrical resistivity measurement of polycrystalline <math>\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}</math> nanosheet stacks.....</b>	<b>183</b>
<b>Figure B.5: Optical microscope images and SEM images of the sample tested for electrical resistivity measurements in Figure 3.2 of the maintext.....</b>	<b>184</b>

<b>Figure B.6: Low (top) to high (bottom) magnification SEM images of the flakes that appear after the autocombustion of the PAA-Metal Complex aqueous solution.....</b>	<b>186</b>
<b>Figure B.7: Samples with identical processing conditions but without kinetic demixing. ....</b>	<b>187</b>
<b>Figure B.8: Low (top) to high (bottom) magnification SEM images of the K-deficient region of the pressurized pellet after the e-field induced kinetic demixing process.....</b>	<b>188</b>
<b>Figure B.9: Optical image of the K-rich portion of the kinetically demixed pellet before and after calcination.....</b>	<b>189</b>
<b>Figure B.10: Low magnification SEM image of a ~2.1 mm long <math>K_xCoO_2 \cdot yH_2O</math> nanosheet stack .....</b>	<b>190</b>
<b>Figure C.1: Low magnification SEM images of the TCO thin film in the maintext Figure 4.1.....</b>	<b>192</b>
<b>Figure C.2: Transparent conductive oxide (TCO) thin films of p-type <math>Ca_3Co_4O_9</math> obtained from viscous resin intermediates with different solute concentrations. ....</b>	<b>193</b>
<b>Figure C.3: Surface SEM images of TCO films made from resins with different solute concentrations.....</b>	<b>194</b>

<b>Figure C.4: Cross-sectional SEM images of the <math>\text{Ca}_3\text{Co}_4\text{O}_9</math> thin film prepared from 4.1 M solute concentration resin after 15 minutes calcination at 650 °C...</b>	<b>195</b>
<b>Figure C.5: SEM images of two <math>\text{Ca}_3\text{Co}_4\text{O}_9</math> thin films that are calcined for (a) 45 and (b) 135 minutes at 650 °C. ....</b>	<b>196</b>
<b>Figure D.1: Phonon transmission measurement through three different types of nanosheet array on the same sample chip.....</b>	<b>202</b>
<b>Figure D.2: Phonon transmission measurement through ten different types of nanosheet array A through J. (See Table 5.1 and Table 5.2 in the main text.)...</b>	<b>203</b>
<b>Figure D.3: Phonon focusing image on (100) plane of Si, generated using MC approach.....</b>	<b>207</b>
<b>Figure D.4: Measured signal produced by phonon transmission at <math>V_s = 2.0</math> mV through un-etched (bulk) Si mesa. ....</b>	<b>210</b>
<b>Figure D.5: Measured signal produced by phonon transmission through ten different arrays of silicon nanosheets for phonon peak frequency of 400 GHz..</b>	<b>211</b>
<b>Figure D.6: Test case performed to validate the MC calculation of phonon surface interaction. ....</b>	<b>212</b>
<b>Figure D.7: Distribution of specularity values predicted by Ziman expression for a 1 nm rough surface interacting with STJ emitted phonons at peak frequency of 400 GHz. ....</b>	<b>213</b>

<b>Figure D.8: Zero-specularity model of transmission through nanosheet arrays, plotted against transmission measurements for peak frequencies 400, 500, 600 GHz. ....</b>	<b>214</b>
<b>Figure D.9: Measurements of phonon transmission through arrays of silicon nanosheets, using phonons of peak frequency 400 GHz. ....</b>	<b>215</b>

## LIST OF TABLES

<b>Table 5.1: Geometry of phonon spectrometers and measured silicon nanosheets.</b>	
.....	118
<b>Table 5.2: Electronic characterization of the phonon spectrometers for silicon nanosheet measurements. ....</b>	123
<b>Table D.1: Results from linear fits of the phonon transmission signal.....</b>	216
<b>Table D.2: The values for elastic constants and density of Si that are used in the phonon focusing algorithm of the MC Approach.....</b>	216
<b>Table D.3: Geometric configurations of phonon sources and detectors in bulk transmission measurements, with corresponding normal state resistance values.</b>	
.....	217
<b>Table D.4: Number of phonons hitting points (A') , (A'') and (A''') in MC simulations of 3 different test cases described in Figure D.6 and their comparison with analytical calculations (See captions of Figure D.6). ....</b>	218

## LIST OF ABBREVIATIONS

AFM – Atomic force microscopy

CHESS - Cornell High Energy Synchrotron Source

CVD - Chemical vapor deposition

E-field – Electric field

ED – Electron diffraction

EDS – Energy dispersive x-ray spectroscopy

EIKD – Electric field induced kinetic demixing

FET - Field emission transistor

FFT – Fast Fourier Transform

FOM – Figure of merit

FT - Fast transverse (polarization)

FWHM - Full width half maximum

HOPG - Highly oriented pyrolytic graphite

HR-TEM – High resolution Transmission electron microscopy

ITO - Indium tin oxide

LA – Longitudinal (polarization)

LCD - Liquid crystal displays

MC – Monte Carlo

OLED - Organic light emitting diodes

PAA - Poly(acrylic acid)

PLD – Pulsed laser deposition

PVD - Physical vapor deposition



SAED – Selected area electron diffraction

SEM - Scanning electron microscope

SGKD - Sol-gel and electric-field induced kinetic-demixing

ST - Slow transverse (polarization)

STJ – Superconducting tunnel junction

TBA - Tetrabutylammonium

TCO - Transparent conducting oxide

TEM – Transmission electron microscopy

VLS - Vapor-liquid-solid

WDS - Wavelength dispersive x-ray spectroscopy

XRD – X-ray diffraction

XPS – X-ray photoelectron spectroscopy

## CHAPTER 1

### 1. INTRODUCTION

#### ***1.0. Nanosheets and Nanostructured Thin Films for Practical Applications***

As the spatial dimensions of a material approaches the nanometer scale, its physical and chemical properties become strongly correlated to the number of atoms or molecules forming the material.<sup>1</sup> In the nanometer size range, continuous energy bands of bulk solids become discrete and their energy band gaps analytically depend on the spatial dimensions of the material.<sup>2</sup> Another typical effect of such size restriction is the strong enhancement in the surface-scattering of heat carriers resulting in significantly reduced mean free path.<sup>3</sup> If a material is restricted along all dimensions, it is called a 0D material; if it has limited size along two dimensions, it becomes a 1D material with a shape similar to a wire; if only one of the dimensions is in the nanoscale, it refers to a 2D material in the form of a sheet.<sup>2</sup>

2D materials, also referred to herein as nanosheets, have been one of the most extensively studied classes of materials within the last decade thanks to the rediscovery of single-atomic-layer graphene in 2004 by Novoselov and Geim.<sup>4</sup> Obtaining single-layer graphene nanosheets is as simple as rubbing HOPG on another surface. These highly transparent ultra-thin sheets are then collected from the surface of an oxidized Si wafer.<sup>1</sup> Extremely mobile electrons in single-layer graphene nanosheets exhibit effectively relativistic behavior with zero effective carrier mass enabling observation of the so called quantum hall effect at room temperature.<sup>5,6</sup> Despite its sub-nanometer thickness, single-layer graphene has very high electronic and thermal conductivity making it ideal for practical applications such as flexible

transparent displays, thermal interface materials and three terminal active circuit elements for ultra-high frequency computer processors.<sup>7-9</sup> Additionally, the valence and conduction band in single-layer graphene nanosheets touch each other at a single point making the band structure of graphene extremely sensitive to external effects such as electric fields, mechanical manipulations and chemical alterations in the surrounding environment.<sup>1</sup> Such characteristic of the graphene band structure makes it very desirable for sensing applications.

Although the majority of research on nanosheets is devoted to graphene, recently there has been numerous studies on nanosheets of other crystalline materials. Motivated by the exfoliation of graphene from HOPG, most of these studies share the same strategy of exfoliating single or few-atom-thick layers from atomically layered bulk solids. This strategy works best for so called Van der Waals solids in which the atomic layers are bound together with weak Van der Waals forces only as in HOPG.<sup>10</sup> The most common methods for making single or few-layer nanosheets from Van der Waals solids are intercalation of molecules or atoms in between the atomic layers of the solid material, mechanical exfoliation using “Scotch tape” and chemical exfoliation by dispersing in a solvent with an appropriate surface tension.<sup>10</sup> Nanosheets of a variety of phases such as MoS<sub>2</sub>, WS<sub>2</sub>, MoSe<sub>2</sub>, MoTe<sub>2</sub>, TaSe<sub>2</sub>, NbSe<sub>2</sub>, NiTe<sub>2</sub>, BN, Bi<sub>2</sub>Te<sub>3</sub>, GeH, SiH<sub>0.5</sub>(OH)<sub>0.5</sub> and Ti<sub>3</sub>C<sub>2</sub>(OH)<sub>2</sub> can be obtained using these methods.<sup>11-14</sup> Among these semiconducting nanosheets, MoS<sub>2</sub> has attracted particular attention due to its practically important properties such as the existence of a direct band gap with strong excitonic effects and the coupling of valley and spin degrees of freedom.<sup>10</sup> MoS<sub>2</sub> monolayers allow field-effect transistors to have very high current

on-off ratios and MoS<sub>2</sub> nanosheets can be used with graphene nanosheets in transistor assemblies.<sup>10</sup>

Metal oxide nanosheets are perhaps the least studied 2D materials. This is probably because atomically layered metal oxide solids have electrostatic interactions in between the atomic layers in addition to Van der Waals force causing exfoliation of metal oxide nanosheets to be more complicated and usually requiring more aggressive chemical methods.<sup>10</sup> In most cases, isolation of single layer metal oxide nanosheets is achieved by the replacement of cations between the atomic layers of the bulk solid with bulky organic cations such as TBA ions.<sup>15</sup> This strategy usually damages or alters the starting compound such that the resulting nanosheets have vastly different physical and chemical characteristic compared to starting ionic solid.<sup>16</sup> Additionally, due to the aggressive top-down chemical route it is unlikely to obtain exfoliated nanosheets that are larger than 10  $\mu\text{m}$ . Nevertheless, some of the metal oxide nanosheets obtained so far are proven to be very promising candidates for electrochemical, electronic, electro-optic and magneto-optics applications.<sup>15,17-19</sup> More details about synthesis and characterization techniques and potential applications for metal oxide nanosheets will be provided in later sections of Chapter 1.

Regardless of the materials type, nanosheet structures also exhibit superior extrinsic properties compared to bulk material, making them ideal for practical devices. Because of their high anisotropy, nanosheets provide the same surface functionality with smaller amounts of material compared to bulk. This enables low material cost, low processing cost, and high packing density, which are desirable, especially in microelectronics and data storage applications. Electromagnetic fields

can penetrate through the nanosheets due to nanoscale thickness leading to efficient operation of field-effect devices and electro-optical devices. Nanosheets dissipate heat easily to the surrounding environment due to large surface area making them suitable for electronic devices with high current density.

In this study, novel fabrication characterization and simulation techniques are presented for inorganic nanosheets and nanostructured thin films with the motivation of advancements in thermal engineering, thermoelectrics, flexible electronics and electro-optics. First, metal oxide nanosheets of  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_{2.y}\text{H}_2\text{O}$  are fabricated using a novel bottom-up method based on sol-gel chemistry, E-field induced kinetic demixing and high temperature heat treatment. Metal oxide nanosheet compounds are chosen from atomically layered oxides since growth of these materials favor nanosheet formation. Another criteria for the nanosheet compounds is having practically desirable properties even in the bulk form so that improvements in the properties through the nanosheet formation can make the material readily applicable to practical devices. Both  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_{2.y}\text{H}_2\text{O}$  are very conductive in bulk form compared to conventional ceramics and suitable candidates for flexible oxide electronics if ductility and flexibility is improved. Bulk  $\text{Na}_x\text{CoO}_2$  is also known to demonstrate superior thermoelectric properties such as high Seebeck coefficient. The thermoelectric efficiency of  $\text{Na}_x\text{CoO}_2$  can be further enhanced due to scattering and confinement of heat carriers in the nanosheet structure. Secondly, novel p-type transparent conducting oxide thin films of  $\text{Ca}_3\text{Co}_4\text{O}_9$  nano-plates are produced. It has been proven in the metal oxide nanosheet synthesis that low temperature calcination of the sol-gel resin produce significantly shorter nanosheets or, in other words, nano-

plates. For the fabrication of the nanostructured p-type TCO thin films, sol-gel process from the metal oxide nanosheet synthesis is combined with spin-coating technique and followed by relatively low temperature heat treatment. It was previously shown that interconnected Ag nanowires can be utilized as solution processed alternatives to n-type TCOs. In these Ag nanowire networks, porosity and nanosize induced reduction in photon-electron interactions increase transparency while high conductivity can still be maintained.<sup>20</sup> In this study, a similar approach is utilized in thin films of  $\text{Ca}_3\text{Co}_4\text{O}_9$  nano-plates in order to produce alternative p-type TCOs which are more challenging to find compared to n-type TCOs. Thirdly, frequency resolved phonon transport experiments are performed on nanofabricated single crystalline Si nanosheets using STJs and experimental results are compared with MC simulations. Nano-science has been proposed to have significant contributions to thermal engineering and thermo-electrics since thermal conductivity can be reduced by orders of magnitude due to phonon confinement and scattering in nanostructures such as nanowires and nanosheets. However, phonon transport in nanostructures, so far, has been investigated through thermal conductance experiments, which provide limited information regarding frequency dependent phonon transmission through the nanostructures. Performing frequency resolved phonon transport measurements through nanostructures will contribute to nanoscale thermal engineering and thermoelectrics. Fabrication and experimental details frequency resolved phonon transport measurements are described elsewhere in detail.<sup>21,22</sup> This study mainly focuses on understanding the frequency resolved phonon transport measurement results using MC simulations and analytical models.

In the rest of Chapter 1, previous studies and background will be provided for metal oxide nanosheets, p-type TCOs and nanoscale phonon transport measurements.

### **1.1. Metal Oxide Nanosheets**

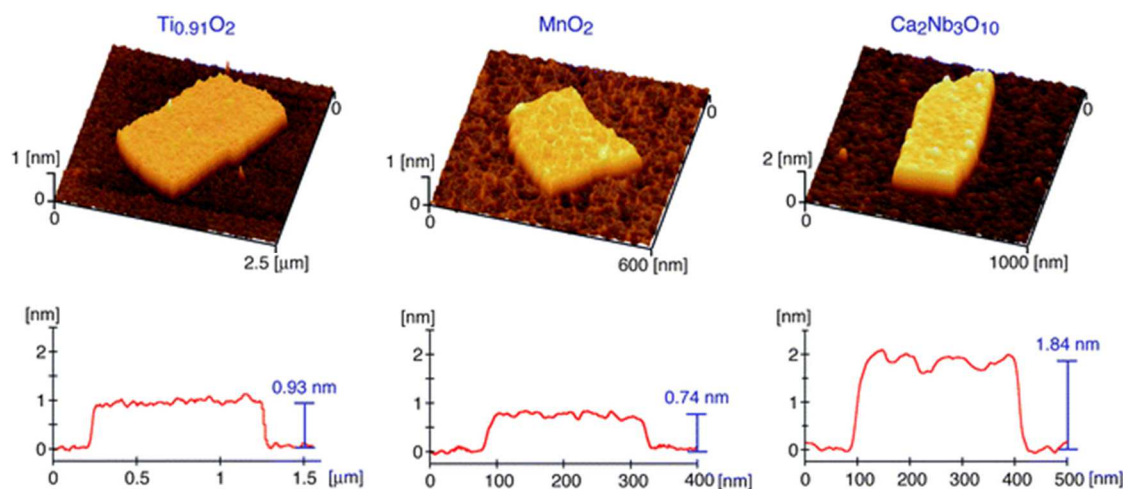
There have been numerous methods to synthesize metal oxide nanosheets that have great potential for practical applications.  $K_{0.15}MnO_2 \cdot 0.43H_2O$ ,  $RuO_{2.1}$ ,  $Ti_{0.91}O_2$  and  $MnO_2$  nanosheets obtained from chemical exfoliation methods exhibit promising electrochemical storage capabilities.<sup>15,23-25</sup> Moreover, chemically exfoliated  $Sr_2Nb_3O_{10}$ ,  $Ti_{0.91}O_2$  and  $Ti_{0.87}O_2$  nanosheets have been shown to function in various active electronic components.<sup>15,26</sup>  $Ga_2O_3$  nanosheets obtained by VLS technique were proposed as ideal materials for dielectric layers in transistors and high-temperature gas sensors.<sup>27</sup> Supercapacitor electrodes of  $NiCo_2O_4$  nanosheets prepared by a template free bottom-up solution growth method provided stable and ultrahigh capacitance.<sup>28</sup>

#### **1.1.0. Synthesis**

##### **1.1.0.0. Chemical Exfoliation**

In chemical exfoliation of metal oxide nanosheets, atomically layered metal oxide with alkali species are delaminated into single layer metal oxide nanosheets. The most well-established method for exfoliating single layer oxide nanosheets is replacement of the alkali cations between the atomic layers of the parent compounds with bulky organic cations such as TBA ions. Initially, atomically layered materials are acid-exchanged into their protonated forms by the removal of the alkali cation.<sup>15</sup> For instance,  $K^+$  ions in  $K_{0.45}MnO_2$  can be replaced by H in HCl solution resulting in  $H_{0.13}MnO_2 \cdot 0.7H_2O$ . The resulting protonic oxides are delaminated through reaction

with a solution containing TBA ions, producing colloidal suspensions of metal oxide nanosheets such as  $\text{MnO}_2$ .<sup>29</sup> The lateral lengths of the exfoliated metal oxide nanosheets are limited with the size of the parent compound crystals.<sup>10</sup> This exfoliation technique has been applied to the exfoliation of many different metal oxide nanosheets with slight variations.<sup>15</sup> **Figure 1.1** shows AFM images and height profiles of  $\text{Ti}_{0.91}\text{O}_2$ ,  $\text{MnO}_2$  and  $\text{Ca}_2\text{Nb}_3\text{O}_{10}$  nanosheets obtained by the chemical exfoliation method (parent compounds:  $\text{Cs}_{0.7}\text{Ti}_{1.825}\text{O}_4$ ,  $\text{K}_{0.45}\text{MnO}_2$ , and  $\text{KCa}_2\text{Nb}_3\text{O}_{10}$  respectively).<sup>15</sup>



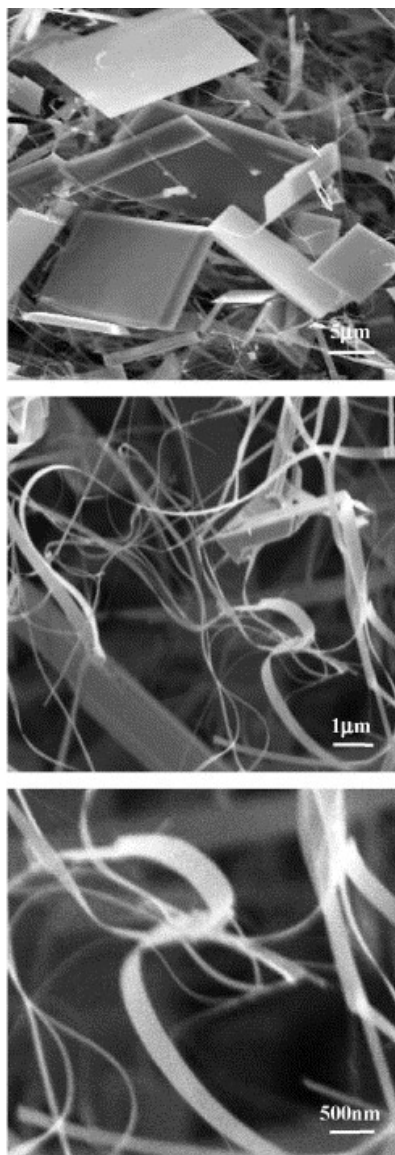
**Figure 1.1: AFM images and height profiles of chemically exfoliated  $\text{Ti}_{0.91}\text{O}_2$ ,  $\text{MnO}_2$  and  $\text{Ca}_2\text{Nb}_3\text{O}_{10}$  nanosheets.** Reproduced from Reference 15 with permission of The Royal Society of Chemistry.

#### **1.1.0.1. Vapor-Liquid-Solid Method**

In a typical VLS synthesis experiment, metallic source material is placed in a tube furnace and an appropriate substrate (e.g. Si wafer) is placed at a certain distance from the source metal. The furnace is then heated to high enough temperatures to melt



and evaporate the metal source. In order to avoid oxidation of metal before evaporation, the tube furnace is kept under continuous Ar flow. Residual amount of oxygen in the tube (likely to originate from air leakage) is expected to provide the source of oxygen for the growth of metal oxide nanosheets and nanobelts.<sup>27</sup> **Figure 1.2** shows SEM image of  $\text{Ga}_2\text{O}_3$  nanosheets and nanobelts obtained with this method.<sup>30</sup>

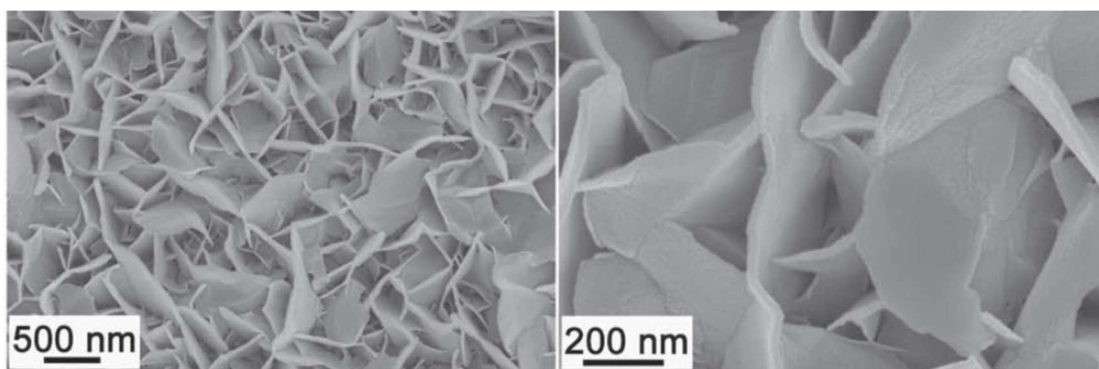


**Figure 1.2: SEM images of  $\text{Ga}_2\text{O}_3$  nanosheets and nanobelts produced by VLS growth.** Adopted from Chemical Physics Letters 378 (5-6), X. Xiang, C. B. Cao, Y.

Guo, and H. S. Zhu, A simple method to synthesize gallium oxide nanosheets and nanobelts, 660-664, Copyright (2003), with permission from Elsevier.

#### **1.1.0.2. Bottom-up Solution Growth Methods**

Although many different solution growth methods have been reported for metal oxide nanosheets, the lateral lengths of the nanosheets in these studies are in sub-micrometer regime making them inappropriate for practical electronic devices. However, several studies reported superior electro-chemical storage capabilities for metal oxide nanosheets produced by solution growth techniques.<sup>31-33</sup> In one of these studies, Co and Ni salts and hexamethylene-tetramine are dissolved in water-ethanol mixture and Ni-Co precursor nanosheets are grown from this solution on a conductive substrate in a heated reaction bottle.<sup>28</sup> The precursor nanosheets are then crystallized into mesoporous  $\text{NiCo}_2\text{O}_4$  nanosheets by a secondary heat treatment. Binder and additive free  $\text{NiCo}_2\text{O}_4$  nanosheet networks (See **Figure 1.3**) exhibit an ultrahigh specific capacitance of  $1626 \text{ F} \cdot \text{g}^{-1}$  even after >3000 cycles at a high charge/discharge current density ( $8.5 \text{ mA} \cdot \text{cm}^{-2}$ ).<sup>28</sup>



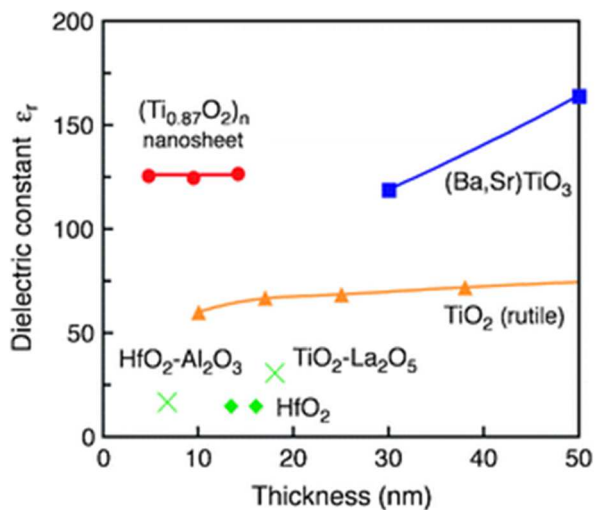
**Figure 1.3: SEM images of  $\text{NiCo}_2\text{O}_4$  precursor nanosheet network.** Adopted from

Advanced Materials, 25 (7), G. Q. Zhang and X. W. Lou, General Solution Growth of

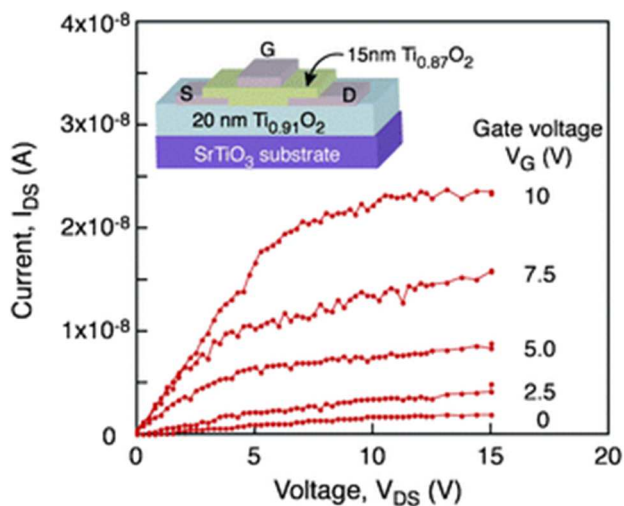
Mesoporous NiCo<sub>2</sub>O<sub>4</sub> Nanosheets on Various Conductive Substrates as High-Performance Electrodes for Supercapacitors, 976-979, Copyright (2013), with permission from Wiley-VCH.

### **1.1.1. Device Integration**

One of the most promising device applications for metal oxide nanosheets is oxide nanoelectronics. Chemically exfoliated nanosheets can function as excellent dielectric gate layers in field effect transistors and semiconductors in p-n junctions.<sup>15,34</sup> Only 5 nm thick Ti<sub>0.87</sub>O<sub>2</sub> nanosheets exhibit very high dielectric constant of  $\epsilon = 125$  with low leakage current density ( $<10^{-7} \text{ A} \cdot \text{cm}^{-2}$ ).<sup>15</sup> **Figure 1.4** shows a plot comparing thickness dependent dielectric constants of Ti<sub>0.87</sub>O<sub>2</sub> nanosheets with other promising dielectric materials.<sup>15</sup> Sr<sub>2</sub>Nb<sub>3</sub>O<sub>10</sub> and Ti<sub>0.91</sub>O<sub>2</sub> nanosheets are shown to be n-type semiconductors.<sup>18,35</sup> Ti<sub>0.87</sub>O<sub>2</sub> dielectric nanosheets can be combined with n-type Ti<sub>0.91</sub>O<sub>2</sub> nanosheets to make FETs entirely based on titania nanosheets with layer-by-layer assembly (see **Figure 1.5** inset drawing). This transistor works in accumulations mode with on/off current ratio of  $10^3$ . Current-voltage (I-V) plot for the titania nanosheet FET device is shown in **Figure 1.5**.<sup>15</sup>



**Figure 1.4: Thickness dependent maximum dielectric constants of various promising dielectric materials compared with dielectric constants of  $\text{Ti}_{0.87}\text{O}_2$  nanosheets.** Reproduced from Reference 15 with permission of The Royal Society of Chemistry.



**Figure 1.5: FET characteristics of titania nanosheet layer-by-layer assembly.** Inset drawing shows device structure. Reproduced from Reference 15 with permission of The Royal Society of Chemistry.

### ***1.2. p-type Transparent Conducting Oxide Thin films***

TCO are essential components of widespread devices such as solar cells, transparent displays, and touch screens. Transparent conducting electrodes in these devices are mostly heavily doped, large band-gap metal oxide semiconductors such as ITO.<sup>36,37</sup> However, all of these commercialized metal oxide semiconductors are n-type, since p-type TCOs, when compared to n-type TCOs, typically have orders of magnitude lower carrier mobility and electrical conductivity.<sup>38-41</sup> Although n-type TCOs meet the necessities of current technology, p-type TCOs can serve as critical components for technological advancements such as p-type counterparts to n-type TCOs in transparent oxide circuits,<sup>40,42</sup> better band matching charge injection layers in organic light emitting devices,<sup>42</sup> and current collectors in solar cells.<sup>43,44</sup> Low cost, easily manufactured p-type TCOs should be readily applicable to near infrared optoelectronics where optical transmission is very poor for n-type TCOs.

So far, the most successful attempts to produce high conductivity p-type TCOs were achieved through CVD and PVD techniques such as metal organic CVD, sputtering, thermal evaporation and PLD.<sup>40,41</sup> RF sputtering grown Mg substituted  $\text{CuCr}_{1-x}\text{Mg}_x\text{O}_2$  thin films have been the most conductive TCOs so far with electrical conductivity of 220 S/cm. Despite relatively high conductivity, visible range optical transparency of these thin films is limited to the visible range with a transparency of 30-40%.<sup>45</sup> Less expensive and more scalable solution-based techniques provide significantly worse electrical conductivity for p-type TCOs compared to PVD-CVD based ones. Spray pyrolysis prepared  $\text{CuCr}_{1-x}\text{Mg}_x\text{O}_2$  thin films are the most conductive solution processed p-type TCO thin films so far with an electrical

conductivity of  $\sim 1$  S/cm.<sup>46</sup> In this study, improvements in solution processed p-type TCO thin films are targeted and more details about existing solution based techniques will be provided in the next two sub-sections of Chapter 1.

### ***1.2.0. Solution-based Synthesis Methods***

#### ***1.2.0.0. Spray Pyrolysis***

In spray pyrolysis technique, a precursor solution with metal salts is atomized to form droplets. These droplets are then sprayed through a heated chamber by pressurize gas flow. The solvent evaporates inside the heated chamber and the remaining solutes are pyrolyzed at the surface of a heated substrate. Sub-micron sized pyrolyzed particles on the substrate form a metal oxide film, and the stoichiometry of the metal oxide compound is dictated by the ratio of metal ion concentrations in the original solution.<sup>47</sup> Thin films produced by spray pyrolysis technique usually have lower quality in comparison to thin films deposited by PVD-CVD.<sup>39</sup> For example, p-type TCO thin films of  $\text{CuCr}_{1-x}\text{Mg}_x\text{O}_2$  produced by spray pyrolysis are more than two orders of magnitude more resistive than RF sputtering prepared ones.<sup>45,46</sup>

#### ***1.2.0.1. Spin Coating***

In the spin coating method, precursor solutions or colloids are applied on the center of a substrate. The substrate can be stationary or spinning during the application of the liquid precursor mixture. For the coating process, the substrate is usually spun at higher speeds to ensure uniform films. The film thickness decreases as the spin rate increases and solution viscosity decreases.<sup>48,49</sup> Most of the liquid content in the original solution usually evaporates during the coating process. Spin coating process is typically followed by heat treatments to evaporate the remaining liquid content and

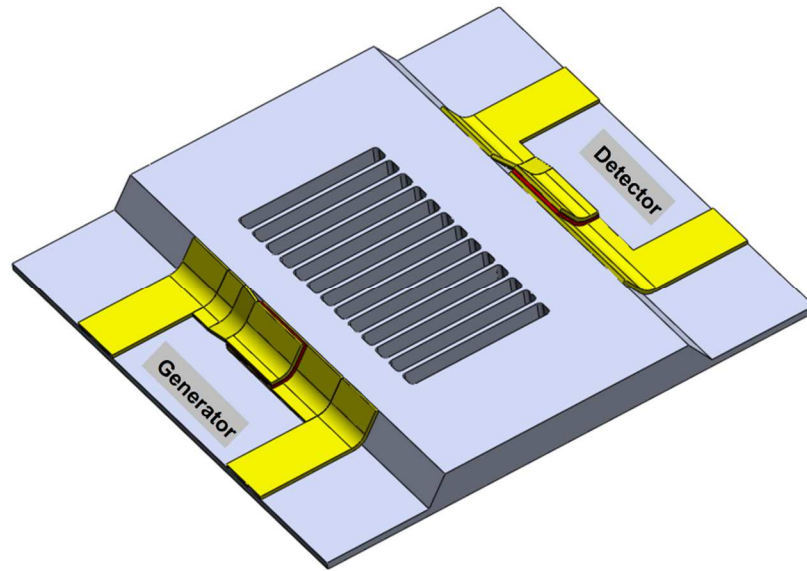
to ensure crystallization of the film.

### ***1.3. Nanoscale Phonon Transport***

Understanding phonon transport in nanostructured materials is essential for future energy applications and microscale thermal engineering. Efficiency of thermoelectric materials can be increased by reducing their thermal conductivity using nanostructures, or nanoscale components in microelectronic circuits can be designed more carefully for efficient cooling.<sup>50-53</sup> However, heat flow at the nanoscale has not been completely understood and so far developing experimental tools for this purpose has been a challenge. Acoustic phonons are the dominant heat carriers in insulators.<sup>54,55</sup> At the nanoscale, as the material dimensions or surface morphology becomes comparable to phonon wavelength, mean free path, and/or coherence length, heat transport will be strongly influenced by phonon confinement and/or phonon-surface scattering.<sup>56,57</sup> In order to completely understand these effects on nanoscale phonon propagation, thermal transport should be measured with a technique that can precisely distinguish different phonon frequencies and tell if the phonons scattered inelastically in transit. Existing methods study thermal transport in nanostructures using Joule-heated metal films.<sup>58-61</sup> However, such thermal conductance measurements employ a broad spectral distribution of phonons and, thus, have very low phonon frequency resolution.

In this study, a nanoscale phonon spectrometry technique with an unprecedentedly high phonon frequency resolution is introduced. The spectrometer relies on STJs to produce and detect non-thermal phonons at frequencies from  $\sim 90$  to  $\sim 870$  GHz. Phonon frequency resolution for this technique is nearly 10 times better

than thermal conductance measurements. Phonon spectrometry experiments performed on Si nanosheets that are microfabricated by etching rectangular trenches into  $0.8\text{ }\mu\text{m}$  high by  $7\text{ }\mu\text{m}$  wide silicon mesa, formed on top of a  $525\text{ }\mu\text{m}$  thick Si wafer. STJ phonon generator and detector are located at the sidewalls of the Si mesa (See **Figure 1.6** for representative drawing of phonon spectrometer device). All of the Si nanosheets in the spectrometry experiments are wider than  $120\text{ nm}$ . Therefore, phonon confinement effects are not expected and nanoscale effects on the phonon propagation should be limited to phonon-surface scattering. In the rest of the subsections in Chapter 1, details regarding phonon spectrometry experimental setup and modeling of phonon spectrometry through nanosheets will be provided.



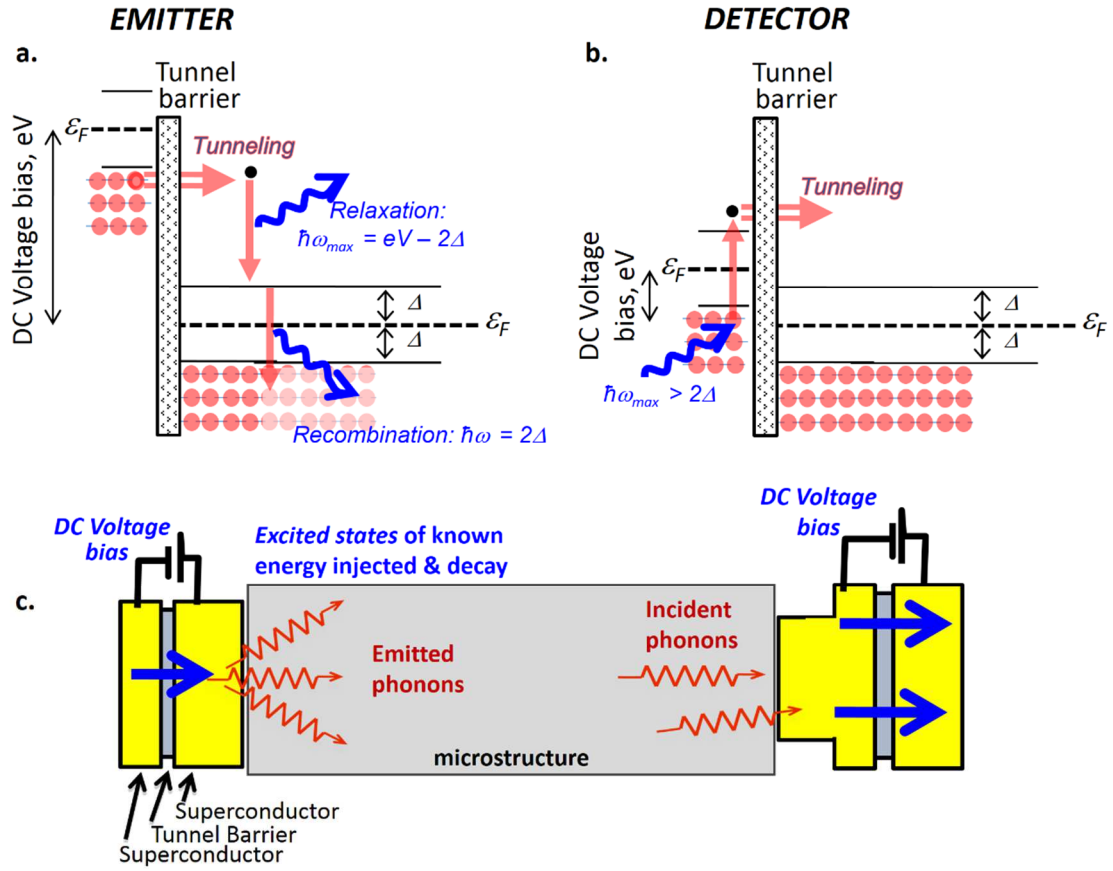
**Figure 1.6: Representative drawing of STJ based microscale phonon spectrometry device.**



### 1.3.0. Phonon Spectrometry Experimental Setup

STJs in phonon emitters and phonon detectors are superconductor-insulator-superconductor junctions consisting of aluminum-aluminum oxide-aluminum layers (Al-Al<sub>x</sub>O<sub>y</sub>-Al). Experiments are performed at ~0.3 K at which temperature Al layers are superconducting.<sup>21</sup> The spectrometer measures the rate of phonons that propagate ballistically through the microstructure between the emitter STJ and detector STJ (See **Figure 1.7c**). Phonons are emitted in STJs by the excitation and decay of quasiparticles (single electrons) in superconducting Al films. When the emitter STJ is DC voltage biased above the superconducting gap ( $2\Delta_e$ ) such that  $V_e \geq 2\Delta_e/e$ , the Cooper pairs in the first aluminum film break into two quasiparticles.<sup>62,63</sup> The quasiparticles tunnel through the insulator barrier into the second Al film at energy states ranging from  $\Delta_e$  to  $eV_e - \Delta_e$ . The quasiparticles in the second Al layer rapidly decay towards the upper edge of the superconducting gap. These phonons emit phonons as they decay with single or multiple steps. Phonons of energies emitted during this relaxation process range from 0 to  $eV_e - 2\Delta_e$  (See **Figure 1.7a** for emitter band diagram).<sup>64</sup> There is a sharp cutoff in relaxation phonon distribution at energy  $eV_e - 2\Delta_e$ . Using this sharp cut-off, a narrow portion of the spectrum that is peaked at energy  $eV_e - 2\Delta_e$  can be isolated by modulation of  $V_e$  (See **Figure 1.8** for simplified representation of emission distribution).<sup>22</sup> Recombination of the quasiparticles into Cooper pairs cause the emission of recombination phonons of energy  $2\Delta_e$ . All relaxation and recombination phonons are emitted and ballistically propagate through the Si mesa; however, only the relaxation phonons are controlled by modulation techniques. For phonon detection, the detector is DC voltage biased in the subgap

regime ( $V_d \sim \Delta_d/e$ ). Phonons arriving the first aluminum layer in the detector junction breaks Cooper pairs and resulting quasiparticles tunnel to the second aluminum layer due to applied voltage bias (See **Figure 1.7b** for detector band diagram).<sup>64</sup> The measured quasiparticle current at the detector constitutes the detector signal which includes both steady state and modulated components. The modulated portion originates from the modulated relaxation phonons arriving at the detector junction. At the emitter junction, modulation of the relaxation phonons is achieved by superimposing AC bias on the DC component which controls the edge frequency in the emission spectrum. Detector signal due to the modulated relaxation phonons is isolated using a lock-in amplifier.<sup>22</sup> (See **Figure 1.9** for simplified phonon transport measurement diagram)



**Figure 1.7: a) and b) Band diagram of quasiparticle tunneling and phonon decay processes in source and detector STJs. c) Diagram showing phonon emission from emitter, propagation through transport medium and detection at the detector.**

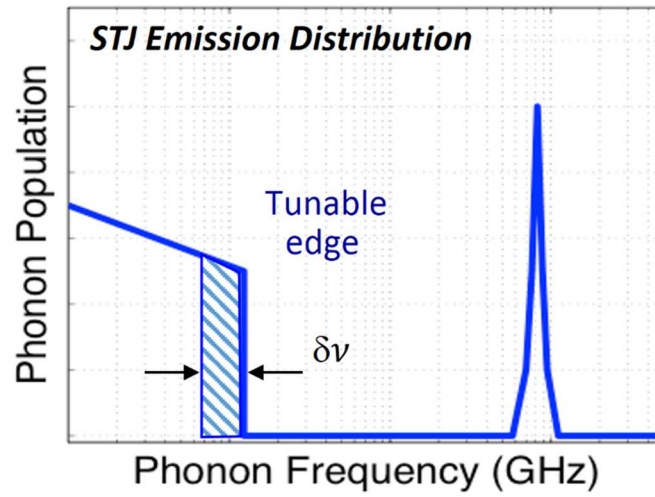


Figure 1.8: Representative diagram for phonon emission

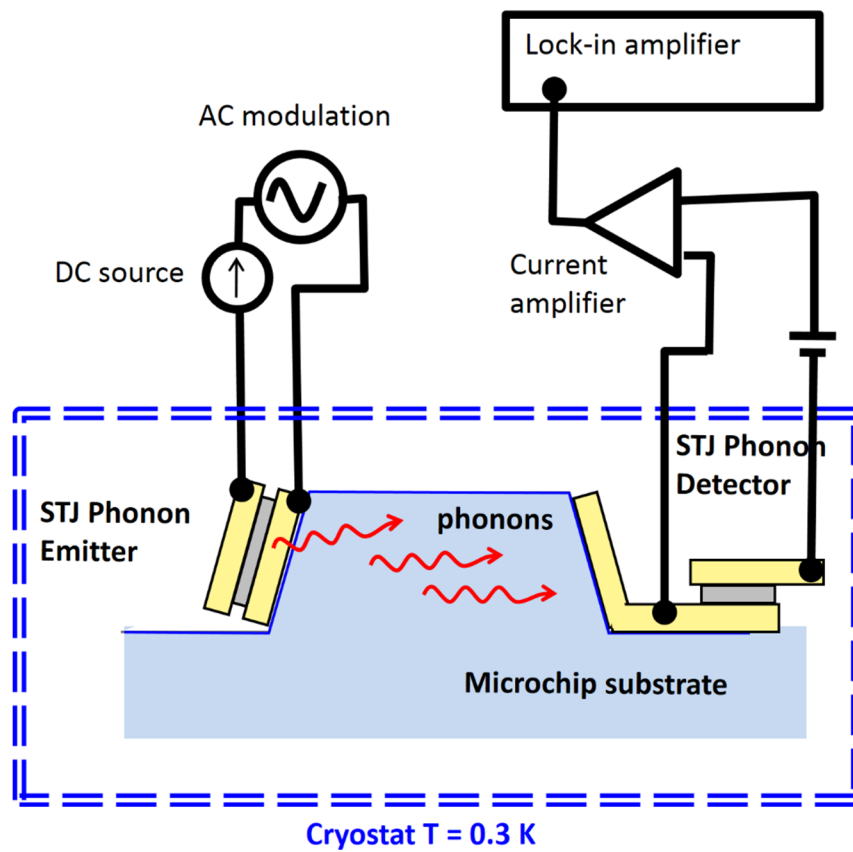


Figure 1.9: Schematic diagram of phonon transport measurement

### ***1.3.1. Modeling of Phonon Spectrometry Through Nanosheets***

Phonon transport through single-crystal silicon nanosheets is modeled by MC simulations. Phonon focusing effects originating from elastic anisotropy of the single crystalline medium is also taken into account. Since phonon-phonon scattering is negligible at low temperatures, only phonon-boundary scattering is taken into account as the source of deviation in the phonon path. After each phonon-surface interaction, the phonons are either scattered diffusively or reflected specularly from the surfaces depending on the specularity of the surfaces. The specularity of the nanosheet surfaces are determined using the well-known Ziman theory based on phonon frequency and surface roughness.<sup>3</sup> After each emission or a diffusive scattering event, the phonon's wavevector is randomized according to a Lambertian distribution. The group velocity vector of the emitted phonons is determined based on the elastic constants of the silicon and the group velocity vectors of the scattered phonons are assumed to be parallel to their wavevectors.

## REFERENCES

- <sup>1</sup> R. Mas-Balleste, C. Gomez-Navarro, J. Gomez-Herrero, and F. Zamora, *Nanoscale* **3** (1), 20 (2011).
- <sup>2</sup> G. A. Ozin, A. C. Arsenault, and L. Cademartiri, *Nanochemistry: A Chemical Approach to Nanomaterials*, 2 ed. (Royal Society of Chemistry, Cambridge, UK, 2009), p.876.
- <sup>3</sup> J. M. Ziman, *Electrons and phonons; the theory of transport phenomena in solids*. (Clarendon Press, Oxford, 1960).
- <sup>4</sup> K. S. Novoselov, A. K. Geim, S. V. Morozov, D. Jiang, Y. Zhang, S. V. Dubonos, I. V. Grigorieva, and A. A. Firsov, *Science* **306** (5696), 666 (2004).
- <sup>5</sup> K. S. Novoselov, Z. Jiang, Y. Zhang, S. V. Morozov, H. L. Stormer, U. Zeitler, J. C. Maan, G. S. Boebinger, P. Kim, and A. K. Geim, *Science* **315** (5817), 1379 (2007).
- <sup>6</sup> Z. Jiang, Y. Zhang, Y. W. Tan, H. L. Stormer, and P. Kim, *Solid State Communications* **143** (1-2), 14 (2007).
- <sup>7</sup> S. Stankovich, D. A. Dikin, G. H. B. Dommett, K. M. Kohlhaas, E. J. Zimney, E. A. Stach, R. D. Piner, S. T. Nguyen, and R. S. Ruoff, *Nature* **442** (7100), 282 (2006).
- <sup>8</sup> K. S. Kim, Y. Zhao, H. Jang, S. Y. Lee, J. M. Kim, K. S. Kim, J. H. Ahn, P. Kim, J. Y. Choi, and B. H. Hong, *Nature* **457** (7230), 706 (2009).
- <sup>9</sup> H. Yang, J. Heo, S. Park, H. J. Song, D. H. Seo, K. E. Byun, P. Kim, I. Yoo, H. J. Chung, and K. Kim, *Science* **336** (6085), 1140 (2012).
- <sup>10</sup> S. Z. Butler, S. M. Hollen, L. Cao, Y. Cui, J. A. Gupta, H. R. Gutierrez, T. F.

- Heinz, S. S. Hong, J. Huang, A. F. Ismach, E. Johnston-Halperin, M. Kuno, V. V. Plashnitsa, R. D. Robinson, R. S. Ruoff, S. Salahuddin, J. Shan, L. Shi, M. G. Spencer, M. Terrones, W. Windl, and J. E. Goldberger, *Acs Nano* **7** (4), 2898 (2013).
- <sup>11</sup> J. N. Coleman, M. Lotya, A. O'Neill, S. D. Bergin, P. J. King, U. Khan, K. Young, A. Gaucher, S. De, R. J. Smith, I. V. Shvets, S. K. Arora, G. Stanton, H. Y. Kim, K. Lee, G. T. Kim, G. S. Duesberg, T. Hallam, J. J. Boland, J. J. Wang, J. F. Donegan, J. C. Grunlan, G. Moriarty, A. Shmeliov, R. J. Nicholls, J. M. Perkins, E. M. Grieveson, K. Theuwissen, D. W. McComb, P. D. Nellist, and V. Nicolosi, *Science* **331** (6017), 568 (2011).
- <sup>12</sup> E. Bianco, S. Butler, S. S. Jiang, O. D. Restrepo, W. Windl, and J. E. Goldberger, *Acs Nano* **7** (5), 4414 (2013).
- <sup>13</sup> H. Okamoto, Y. Sugiyama, and H. Nakano, *Chemistry-a European Journal* **17** (36), 9864 (2011).
- <sup>14</sup> M. Naguib, M. Kurtoglu, V. Presser, J. Lu, J. J. Niu, M. Heon, L. Hultman, Y. Gogotsi, and M. W. Barsoum, *Adv. Mater.* **23** (37), 4248 (2011).
- <sup>15</sup> M. Osada and T. Sasaki, *J. Mater. Chem.* **19** (17), 2503 (2009).
- <sup>16</sup> Y. Masuda, Y. Hamada, W. S. Seo, and K. Koumoto, *J. Nanosci. Nanotechnol.* **6** (6), 1632 (2006).
- <sup>17</sup> N. Sakai, Y. Ebina, K. Takada, and T. Sasaki, *J Electrochem Soc* **152** (12), E384 (2005).
- <sup>18</sup> H. Sato, K. Okamoto, K. Tamura, H. Yamada, K. Saruwatari, T. Kogure, and A. Yamagishi, *Applied Physics Express* **1** (3), 35001 (2008).

- <sup>19</sup> J. Sato, H. Kato, M. Kimura, K. Fukuda, and W. Sugimoto, *Langmuir* **26** (23), 18049 (2010).
- <sup>20</sup> J. Y. Lee, S. T. Connor, Y. Cui, and P. Peumans, *Nano Lett.* **8** (2), 689 (2008).
- <sup>21</sup> O. O. Otelaja, J. B. Hertzberg, M. Aksit, and R. D. Robinson, *New J. Phys.* **15**, 43018 (2013).
- <sup>22</sup> J. B. Hertzberg, O. O. Otelaja, N. J. Yoshida, and R. D. Robinson, *Rev. Sci. Instrum.* **82** (10), 104905 (2011).
- <sup>23</sup> M. Yeager, W. X. Du, R. Si, D. Su, N. Marinkovic, and X. W. Teng, *J. Phys. Chem. C* **116** (38), 20173 (2012).
- <sup>24</sup> W. Sugimoto, H. Iwata, Y. Yasunaga, Y. Murakami, and Y. Takasu, *Angewandte Chemie-International Edition* **42** (34), 4092 (2003).
- <sup>25</sup> L. Z. Wang, K. Takada, A. Kajiyama, M. Onoda, Y. Michiue, L. Q. Zhang, M. Watanabe, and T. Sasaki, *Chem Mater* **15** (23), 4508 (2003).
- <sup>26</sup> M. Osada, K. Akatsuka, Y. Ebina, H. Funakubo, K. Ono, K. Takada, and T. Sasaki, *Acs Nano* **4** (9), 5225 (2010).
- <sup>27</sup> Z. R. Dai, Z. W. Pan, and Z. L. Wang, *J. Phys. Chem. B* **106** (5), 902 (2002).
- <sup>28</sup> G. Q. Zhang and X. W. Lou, *Adv. Mater.* **25** (7), 976 (2013).
- <sup>29</sup> Y. Omomo, T. Sasaki, L. Z. Wang, and M. Watanabe, *J Am Chem Soc* **125** (12), 3568 (2003).
- <sup>30</sup> X. Xiang, C. B. Cao, Y. Guo, and H. S. Zhu, *Chemical Physics Letters* **378** (5-6), 660 (2003).
- <sup>31</sup> L. H. Wu, H. B. Yao, B. Hu, and S. H. Yu, *Chem Mater* **23** (17), 3946 (2011).
- <sup>32</sup> C. Z. Yuan, J. Y. Li, L. R. Hou, L. Yang, L. F. Shen, and X. G. Zhang,



- Electrochimica Acta **78**, 532 (2012).
- <sup>33</sup> Q. Yang, Z. Y. Lu, X. M. Sun, and J. F. Liu, Sci Rep-Uk **3** (2013).
- <sup>34</sup> M. Osada and T. Sasaki, Adv. Mater. **24** (2), 210 (2012).
- <sup>35</sup> N. Sakai, Y. Ebina, K. Takada, and T. Sasaki, J Am Chem Soc **126** (18), 5851 (2004).
- <sup>36</sup> D. S. Hecht, L. B. Hu, and G. Irvin, Adv. Mater. **23** (13), 1482 (2011).
- <sup>37</sup> D. Ginley, H. Hosono, and D.C. Paine, *Handbook of Transparent Conductors*. (Springer, 2010).
- <sup>38</sup> G. Hautier, A. Miglio, G. Ceder, G. M. Rignanese, and X. Gonze, Nat. Commun. **4** (2013).
- <sup>39</sup> E. Arca, K. Fleischer, and I. V. Shvets, Appl. Phys. Lett. **99** (11) (2011).
- <sup>40</sup> A. N. Banerjee and K. K. Chattopadhyay, Prog. Cryst. Growth. Ch. **50** (1-3), 52 (2005).
- <sup>41</sup> S. Sheng, G. J. Fang, C. Li, S. Xu, and X. Z. Zhao, Phys. Status Solidi A- Appl. Mat. **203** (8), 1891 (2006).
- <sup>42</sup> E. Fortunato, P. Barquinha, and R. Martins, Adv. Mater. **24** (22), 2945 (2012).
- <sup>43</sup> W. Beyer, J. Hupkes, and H. Stiebig, Thin Solid Films **516** (2-4), 147 (2007).
- <sup>44</sup> H. Y. Liu, V. Avrutin, N. Izyumskaya, U. Ozgur, and H. Morkoc, Superlattices Microstruct. **48** (5), 458 (2010).
- <sup>45</sup> R. Nagarajan, A. D. Draeseke, A. W. Sleight, and J. Tate, J. Appl. Phys. **89** (12), 8022 (2001).
- <sup>46</sup> S. H. Lim, S. Desu, and A. C. Rastogi, J. Phys. Chem. Solids **69** (8), 2047 (2008).

- 47 G. J. Exarhos and X. D. Zhou, *Thin Solid Films* **515** (18), 7025 (2007).
- 48 A. Aegerter, *Sol-Gel Technologies for Glass Producers and Users*. (Springer, 2004).
- 49 A. Busnaina, *Nanomanufacturing Handbook*. (Taylor & Francis, 2010).
- 50 A. I. Hochbaum, R. K. Chen, R. D. Delgado, W. J. Liang, E. C. Garnett, M. Najarian, A. Majumdar, and P. D. Yang, *Nature* **451** (7175), 163 (2008).
- 51 A. Balandin and K. L. Wang, *Phys. Rev. B* **58** (3), 1544 (1998).
- 52 A. Shakouri, *Proceedings of the IEEE* **94** (8), 1613 (2006).
- 53 Christopher J. Vineis, Ali Shakouri, Arun Majumdar, and Mercouri G. Kanatzidis, *Adv. Mater.* **22** (36), 3970 (2010).
- 54 J. Zou and A. Balandin, *J. Appl. Phys.* **89** (5), 2932 (2001).
- 55 D. G. Cahill, W. K. Ford, K. E. Goodson, G. D. Mahan, A. Majumdar, H. J. Maris, R. Merlin, and S. R. Phillpot, *J. Appl. Phys.* **93** (2), 793 (2003).
- 56 J. K. Yu, S. Mitrovic, D. Tham, J. Varghese, and J. R. Heath, *Nat. Nanotechnol.* **5** (10), 718 (2010).
- 57 G. Chen, *Nanoscale Energy Transport and Conversion*. (Oxford Uni. Press, New York, 2005), p.531.
- 58 J. S. Heron, T. Fournier, N. Mingo, and O. Bourgeois, *Nano Lett.* **9** (5), 1861 (2009).
- 59 K. Schwab, E. A. Henriksen, J. M. Worlock, and M. L. Roukes, *Nature* **404** (6781), 974 (2000).
- 60 K. Hippalgaonkar, B. L. Huang, R. K. Chen, K. Sawyer, P. Ercius, and A. Majumdar, *Nano Lett.* **10** (11), 4341 (2010).

- <sup>61</sup> J. H. Seol, I. Jo, A. L. Moore, L. Lindsay, Z. H. Aitken, M. T. Pettes, X. S. Li, Z. Yao, R. Huang, D. Broido, N. Mingo, R. S. Ruoff, and L. Shi, *Science* **328** (5975), 213 (2010).
- <sup>62</sup> W. Eisenmenger and A. H. Dayem, *Phys. Rev. Lett.* **18** (4), 125 (1967).
- <sup>63</sup> H. Kinder, K. Laszmann, and W. Eisenmenger, *Physics Letters A* **31** (8), 475 (1970).
- <sup>64</sup> W. Eisenmenger, in *Physical Acoustics*, edited by Warren P. Mason and R. N. Thurston (Academic Press, New York, 1976), Vol. 12, pp. 79

## CHAPTER 2

### 2. SCALABLE NANOMANUFACTURING OF MILLIMETER-LENGTH 2D

#### $\text{Na}_x\text{CoO}_2$ NANOSHEETS \*

##### **2.0. Abstract**

A novel, scalable nanomanufacturing technique is reported for batch fabrication of nanoscale-thick  $\text{Na}_{0.7}\text{CoO}_2$  nanosheets. The nanomanufacturing technique is a high-yield, bottom-up process that is capable of producing tens of thousands of nanosheets stacked into a macro-scale pellet. The nanosheets are uniform in length and shape with very high crystal anisotropy. The nanosheet thicknesses can be 10–100 nm while their lengths can measure up to 1.8 mm long. The typical dimension ratios are highly anisotropic, at  $10^{-5}$ :1:1 (thickness:length:width). X-ray synchrotron studies indicate that the 2D crystals are stacked in a turbostratic arrangement with rotational misalignment with respect to the stacking axis. The stacked nanosheets are readily delaminated into very large ( $350\text{ }\mu\text{m}$  x  $150\text{ }\mu\text{m}$  x 100 nm) free-standing 2D crystals. The novel nanomanufacturing technique is based on sol-gel and electric-field induced kinetic-demixing followed by a brief high temperature treatment, thus providing an efficient means of large scale crystal growth requiring only a simple furnace and power supply. Evidence shows that the demixing process increases the concentration of Na ions and that demixing is necessary to produce the millimeter-length nanosheets. EIKD is successfully

---

\* Originally Published as: Mahmut Aksit, David P. Toledo and Richard D. Robinson, “Scalable nanomanufacturing of millimetre-length 2D  $\text{Na}_x\text{CoO}_2$  nanosheets”, *Journal of Materials Chemistry* 22, 5936 (2012). Reprinted with Permission from The Royal Society of Chemistry.

performed at low temperatures ( $<300\text{ }^{\circ}\text{C}$ ), which is more than three times lower than past kinetic-demixing temperatures.<sup>1</sup>

### **2.1. Introduction**

Two-dimensional (2D) oxide crystals present exciting opportunities for new scientific and technological breakthroughs with their novel electronic, ferromagnetic, magneto-optical, electrochemical, catalytic, and photoresponsive properties.<sup>2-6</sup>

Although graphene has been well studied, the synthesis of free-standing 2D nanosheet materials still constitutes a largely unexplored area of nanoscience, especially with inorganic compounds such as transition metal oxides.<sup>2-4,7</sup>

2D metal oxide nanostructures have great potential for practical applications. Nanosheets of  $\text{RuO}_{2.1}$  indicated high performance as electro-chemical supercapacitors.<sup>8</sup> Reassembled  $\text{Ti}_{0.91}\text{O}_2$  and  $\text{MnO}_2$  nanosheets showed electrochemical capacities comparable to Li-Ion batteries.<sup>6,9,10</sup> N-type photo-semiconducting nanosheets of  $\text{Sr}_2\text{Nb}_3\text{O}_{10}$  were recently shown to have diode type characteristics in nanosheet p-n junctions.<sup>11</sup>  $\text{Ti}_{0.91}\text{O}_2$  and  $\text{Ti}_{0.87}\text{O}_2$  nanosheets were used, respectively, as semi-conducting channels and gate dielectric layers in FET devices.<sup>2</sup> Gigantic magneto-optical effects were observed in multilayer assemblies of two-dimensional  $\text{Ti}_{0.8}\text{Co}_{0.2}\text{O}_2$  and  $\text{Ti}_{0.6}\text{Fe}_{0.4}\text{O}_2$  nanosheets.<sup>12</sup>

One of the most exciting new applications for metal oxides is thermoelectrics. Thermoelectric oxides are chemically and thermally stable. They can be composed of nontoxic, light, cheap, and naturally abundant elements. They can be produced through environmentally friendly means, and are expected to play a vital role in extensive applications for waste heat recovery under atmospheric conditions.<sup>13</sup> Most

work on thermoelectric oxides has been devoted to bulk single crystals and polycrystalline samples. Very little work has been pursued for nanocrystalline oxide thermoelectrics, except for thin films on substrates.<sup>14</sup> Scaling these materials to nanoscale dimensions (<100 nm) offers additional control, such as decreasing the phonon thermal conductivity by phonon confinement<sup>15</sup>, which is an effective means of increasing thermoelectric figure of merit (ZT).

Recently, the complex metal oxide  $\text{Na}_x\text{CoO}_2$  has been vigorously researched due to its potential in thermoelectric energy harvesting.<sup>16,17</sup>  $\text{Na}_x\text{CoO}_2$  is composed of two-dimensional triangular  $\text{CoO}_2$  layers with  $\text{Na}^+$  intercalated between the layers, occupying a variety of possible sites.<sup>18,19</sup> The  $x=0.7$  phase ( $\text{Na}_{0.7}\text{CoO}_2$ ) is a surprisingly good thermoelectric material with metallic conductivity ( $200 \mu\Omega\cdot\text{cm}$  at 300 K), a large Seebeck Coefficient ( $100 \mu\text{V/K}$  at 300 K), and a thermoelectric power factor as high as the industry standard  $\text{Bi}_2\text{Te}_3$ .<sup>16,20</sup> Scaling  $\text{Na}_x\text{CoO}_2$  to nanometer sizes will allow this important energy material to be optimized through size-dependent property engineering.

Chemical exfoliation of bulk materials has been the most common method to produce oxide nanosheets<sup>2,4</sup>, and is responsible for the majority of exiting advances in this field. Unfortunately, this aggressive top-down chemical process can damage or alter the starting compound. For instance, chemical exfoliation of thermoelectric  $\text{Na}_x\text{CoO}_2$  layers yields  $\text{CoO}$  nanosheets<sup>21</sup>, which are non-metallic and not useful thermoelectric materials. Chemical exfoliation techniques, which delaminate layers from bulk samples, generally produce a low yield of sheets with short lateral lengths (typically < 10  $\mu\text{m}$ ).<sup>4</sup>

In this work, a scalable nanomanufacturing technique for batch fabrication of 2D  $\text{Na}_x\text{CoO}_2$  nanosheets is reported. The method is based on sol-gel chemistry and kinetic-demixing. This bottom-up synthesis is capable of producing tens of thousands of well-defined nanosheets stacked into a macro-sized pellet, providing an efficient means of large-scale crystal growth as compared to conventional nanofabrication and crystal growth techniques. The nanosheets are uniform in length and shape with high aspect ratios of nanometer thickness and millimeter lateral lengths ( $10^{-5}$ :1:1). They are readily delaminated into free-standing nanosheets without changing crystal structure.

Our method of producing millimeter length 2D nanosheets of  $\text{Na}_{0.7}\text{CoO}_2$  will facilitate this important material for the nano-size property enhancements in potential thermoelectric devices. Scaling  $\text{Na}_x\text{CoO}_2$  to nanometer dimensions is expected to significantly improve its thermoelectric performance due to phonon scattering-confinement effects.<sup>15</sup> Phonon scattering-confinement by the nanometer dimensions is expected to reduce the thermal conductivity of  $\text{Na}_{0.7}\text{CoO}_2$ , which should result in enhanced ZT. In addition, due to the millimeter scale lateral lengths, the  $\text{Na}_x\text{CoO}_2$  nanosheets can be readily applied to practical macroscopic thermoelectric devices.

## **2.2. Experimental**

### **2.2.0. Synthesis of $\text{Na}_{0.7}\text{CoO}_2$ nanosheets**

An aqueous solution is prepared at room temperature by mixing appropriate quantities of PAA (average molecular weight:  $M_w \sim 1800$ ), cobalt(II) nitrate hexahydrate (0.230 M) and sodium nitrate (0.165 M) in de-ionized water. The Na to Co ratio is set to 0.72. The ratio of PAA carboxylate groups to total metal ions is 1:2.

The solution is stirred and evaporated at 150 °C on a hotplate until it reached 20% of the initial volume. The resulting dark red solution is then autocombusted at 500 °C. The resulting black powder is uni-axially pressed into pellets with a rectangular die set at 400 MPa. A typical size of the pellet is 10x6x5 mm (LxWxT). The pellet is kinetically-demixed at < 300 °C with a constant-current electrical field of 500 mA through Cu plates and contacts made of silver epoxy. Over the course of 48 hours the voltage fluctuates between 20 V and 5 V, and decreases towards the end of the process. During the kinetic-demixing process, a red emitting current pathway appears on the pellet with temporally changing position. After the kinetic-demixing, the pellet separates into Na-rich and Na-depleted regions and the boundary between the two regions is clearly distinguishable. The Na-depleted region is mechanically weak, porous and grey while the Na-rich region is mechanically strong, dense and black. The Na-rich region of the pellet is separated and calcined in a tube furnace to finally obtain  $\text{Na}_x\text{CoO}_2$  nanosheets. The furnace is rapidly heated to 1030 °C and held for 15 minutes. The temperature is then ramped down to 1000 °C in 1.5 hours and down to room temperature in the following 3 hours. After calcination,  $\text{Na}_x\text{CoO}_2$  phases have high Na content ( $0.8 < x < 0.93$ ). To obtain the thermoelectric phase ( $\text{Na}_{0.7}\text{CoO}_2$ ) the sample is subjected to a secondary heat treatment at 850 °C for ~30 hours.

### ***2.2.1. Exfoliation of the nanosheets***

Exfoliation of the  $\text{Na}_{0.7}\text{CoO}_2$  nanosheets is accomplished through thermal shock and ultrasonication. Pellets of stacked  $\text{Na}_x\text{CoO}_2$  nanosheets are repeatedly quenched from 500°C in clean (18 MΩ), room temperature water, followed by a brief sonication in water. The nanosheets are then collected from the water surface.



### **2.2.2. XRD, SEM, EDS and TEM characterization**

All of the intermediate and final products of the synthesis procedure were examined by a high resolution ( $\sim 1$  nm at 2 keV) SEM (LEO 1550 FESEM), and by XRD. At least 100 nanosheets were measured by SEM for nanosheet thickness statistical analysis. SEM-EDS was performed for compositional analysis. Conventional  $2\theta$ - $\theta$  XRD measurements were conducted using a Scintag (Pad-X) with a Cu  $K\alpha 1$  source. The samples were finely ground before the XRD based phase analysis measurements in order to minimize possible texturing and crystal alignment effects. The measured XRD peak locations were identical to those of un-ground samples. TEM, dark field imaging, and ED were performed with an FEI T12 Twin TEM.

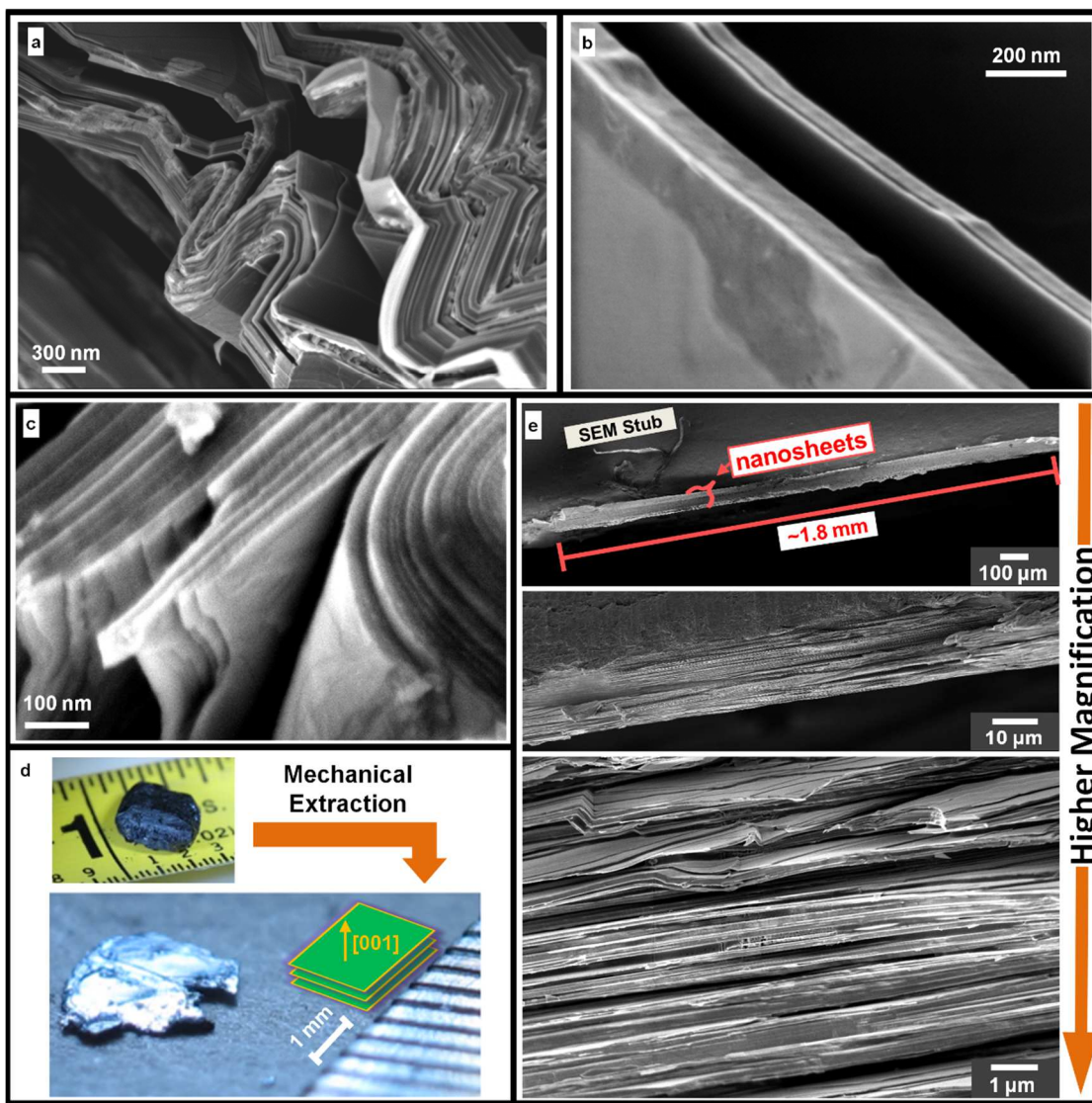
### **2.2.3. Synchrotron x-ray analysis**

To effectively characterize the obtained stacked nanosheet structures a 6-axis diffractometer at CHESS was used. The independent 6-axis allows for 2D scans in reciprocal space which can be arranged to obtain reciprocal lattice mapping, rocking curve, and combinations of these two.<sup>22</sup> The nanosheets were oriented with their [001] axis aligned vertically, and the incident beam skimmed the horizontal plane ( $\sim 0.05^\circ$ ). A linear detector (represented by  $\Delta$  axis in angular space) was fixed to a position defined with respect to the Bragg Condition of  $\{101\}$  planes of  $\text{Na}_{0.7}\text{CoO}_2$  and aligned parallel to [001]. During the measurements, the sample was rotated about [001] ( $\phi$  rotational axis). This measurement method simultaneously provided a rocking curve type scan for (100) planes due to the  $\phi$  rotation and a  $\theta$ - $2\theta$  type of scan for (001) planes due to the linear range of the detector. Additional details of this process are

provided in **Figure A.5** and **A.6** in Appendix A.

### **2.3. Results and Discussion**

SEM images of the metal oxide nanosheets are shown in **Figure 2.1**. The nanosheets lengths are millimeters (**Figure 2.1d** and **Figure A.1**, Appendix A, show two different  $\sim 1.8$  mm-long stacks of nanosheets), can be easily bent (**Figure 2.1a**), and have a smooth surface (**Figure 2.1b**). According to the statistical analysis performed by SEM, the nanosheet thickness is 18.2 nm in average (**Figure 2.1c** and **Figure A.2**, Appendix A). The nanosheets are produced as stacks of sheets (**Figure 2.1e**) that can number in the tens to hundreds of thousands of nanosheets, per stack. Optical images of the bulk product are shown in **Figure 2.1d**, along with the extraction of a stack of nanosheets.



**Figure 2.1: SEM and optical microscope images of metal oxide nanosheets: (a)**

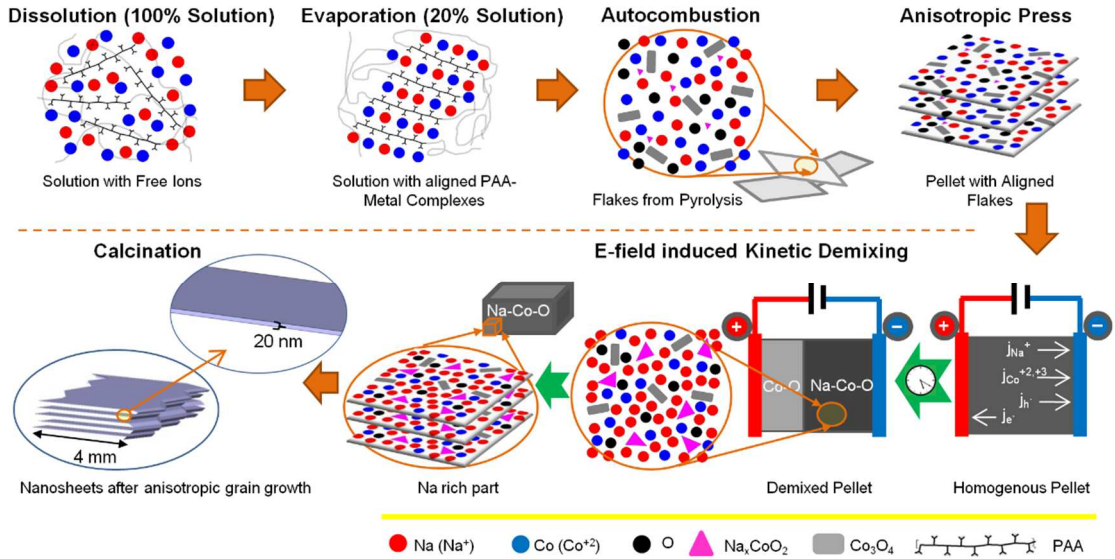
SEM image of metal oxide nanosheets displaying extreme ductility after mechanical bending is applied. (b) SEM image of two individual nanosheets of  $\text{Na}_{0.7}\text{CoO}_2$  showing thicknesses of  $\sim 50 - 100$  nm. (c) SEM image of the nanosheets showing the typical nanosheet thickness profile (18.2 nm in average, See **Figure A.2**, Appendix A). (d)(top) Optical image of the bulk pellet product consisting of thousands of stacked nanosheets and (bottom) a single nanosheet stack obtained by mechanical extraction. The inset drawing shows the alignment of the nanosheets in the nanosheet

stack. (e) SEM images of nanosheet stacks from low magnification (top) to high magnification (bottom). Total nanosheet length is 1.8 millimeters, and stack thickness is nominally 100 microns.

The synthetic procedure for our  $\text{Na}_x\text{CoO}_2$  nanosheets consists of a) the Pechini-method coordination of metal ions, b) pyrolysis into oxide flakes, c) pressurized pellet formation, d) electric-field (E-field) induced kinetic-demixing, and e) calcination (**Figure 2.2**).

Our sol-gel synthesis employs the Pechini method where metal precursors are dissolved in an appropriate solvent by an organic complexing agent, and the resulting homogenous liquid solution is evaporated into a viscous resin intermediate.<sup>23</sup> The resin is then heated to autocombustion. The organic components are pyrolyzed in this process, and the resulting product is a uniform metal oxide mixture. The molar ratio between the different metal cations in the starting solution is maintained in the oxide mixture.<sup>23</sup> By proper choice of the solvent, organic complexing agent and metal precursors in the liquid solution, the atomic homogeneity can be maintained in the metal oxide autocombusted product.<sup>24</sup> In this work, the Pechini method is modified to entrap the metal ions in solutions by using PAA as the chelating agent. The metal cations are expected to be stabilized by the chelating groups on the ligand, through dipole forces between water molecules and metal ions, and by the crosslinking and physical tangling of polymer chains which can trap both the cations and solvents in a drying polymer-metal ion complex sheet, termed “polymeric entrapment” or “steric entrapment”.<sup>25,26</sup> This polymer-metal ion complex sheet is not to be confused by

either atomic layering of  $\text{Na}_x\text{CoO}_2$  or the 2D  $\text{Na}_x\text{CoO}_2$  crystals reported in this work. The crosslinking is likely to occur bi-axially which causes parallel alignment of PAA-metal complexes, similar to that reported for alginates.<sup>27</sup> The structure of these polymeric entrapped layers is evident after the solution is pyrolyzed. SEM images show <200 nm thick and <400  $\mu\text{m}$  long autocombusted flakes which include  $\text{Co}_3\text{O}_4$  crystals with a relatively minor amount of CoO according to the XRD analysis (**Figure 2.3a** and **Figure A.3**, Appendix A). Additionally, edge-to-edge-connected plate-like particles form after the calcination of the autocombusted powder (650 °C, 1 hr) (**Figure 2.3b**). Similar behavior was observed by Zhang et. al.<sup>28</sup> for alginate gel based synthesis of tiled  $\text{Na}_x\text{Co}_2\text{O}_4$  crystals. The entrapment and crosslinking by PAA is crucial for forming the autocombusted flakes: if citric acid is used in place of PAA and the ratio of carboxylic moieties to the number of metal cations is fixed (2:1), flake-like formations do not occur after autocombustion and the volume of the autocombusted form is much smaller compared to the case with PAA (i.e., less than 1/3). Similarly, flake-like formations do not form when the ratio of PAA coordination sites to metal cations is varied from the critical value of 2. When the ratio is 1 (one carboxylic site to one metal cation) an explosive combustion occurs, and the process does not produce flake-like formations. When the number of carboxylic sites exceed the number of metal cations (i.e., ratio of 3 or 4), the autocombustion is slower, the volume expansion during the autocombustion is significantly reduced and flake-like formations are not produced.



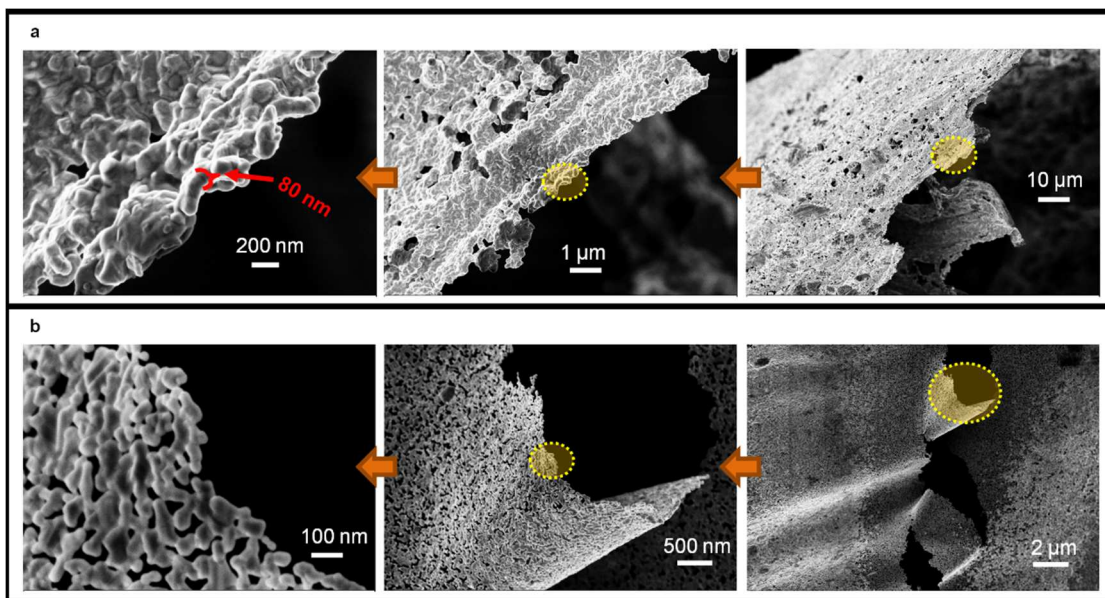
**Figure 2.2: Schematic diagram for the synthesis procedure for millimeter-length**

**2D Na<sub>x</sub>CoO<sub>2</sub> nanosheets:** Metal precursors and PAA are dissolved in water. The solution is evaporated to eighty-percent of its initial volume, leading to metal complex formation. The viscous liquid is autocombusted into a pyrolyzed phase, which form as flakes. The pyrolyzed flakes are uniaxially pressed into a rectangular pellet. E-field is applied to the pellet, and kinetic-demixing occurs due to the different mobility of Na and Co atoms. The Na-rich region is calcined at 1030 °C leading to the formation of the nano-layered structure due to anisotropic grain growth.

After the autocombusted flakes are pressurized by a rectangular die, XRD results show that (1) the cobalt oxide phases formed during the autocombustion are still present, and (2) the emergence of a Na<sub>x</sub>CoO<sub>2</sub> phase (**Figure A.3**, Appendix A). Similar to the autocombusted product, the pressurized pellet includes Co<sub>3</sub>O<sub>4</sub> crystals with relatively minor amount of CoO. However, after the pressurization process a distinct and relatively low intensity peak appears which matches with Na<sub>0.6</sub>CoO<sub>2</sub> phase

(**Figure A.3**, Appendix A). All of the peak intensities for both the autocombusted flakes and the pressurized pellet XRD were low which can be an indication of partially amorphous structure.

The next step, kinetic-demixing, is the critical component to forming millimeter-length nanosheets. After application of a 500 mA current and a 1030 °C calcination, the nanosheet lateral lengths can reach ~1.8 millimeters. Samples with identical processing conditions, but *without* kinetic-demixing, reach only up to ~200  $\mu\text{m}$ , i.e., nine-times shorter. The alignment of the pressurized flakes has an influence on the kinetic-demixing: E-field induced kinetic-demixing is only possible perpendicular to the pressure axis. When an E-field is applied parallel to the pressure axis no current is observed (under identical voltage conditions as the perpendicular arrangement). This directional limitation on the E-field induced kinetic-demixing is likely a result of the lateral alignment of the autocombusted flakes after the uni-axial pressure process; the flakes provide a more contiguous pathway for the current than a cross-plane pathway provides.



**Figure 2.3: (a) SEM images of the flakes that appear after the autocombustion of the PAA-Metal Complex aqueous solution. The thickness of the flakes is < 200 nm and the length measures up to 400  $\mu\text{m}$ . (b) SEM images of the edge-to-edge-connected  $\text{Na}_x\text{CoO}_2$  crystals that appear after the low temperature calcination (650  $^\circ\text{C}$ ) of the autocombusted form. SEM images are sorted from high magnification to low magnification (left to right). The magnified regions are circled in the low magnification images.**

After the application of the E-field to the homogenous pellet, an abrupt transition in Na concentration is observed along the E-field application axis and, according to XRD results, new phases form in the Na-rich part of the pellet. Elemental analysis from an SEM-EDS line scan shows the abrupt transition in Na content and morphology between the sample regions adjacent to the cathode and that to the anode (**Figure 2.4**). The region near the anode is Na-depleted and extremely

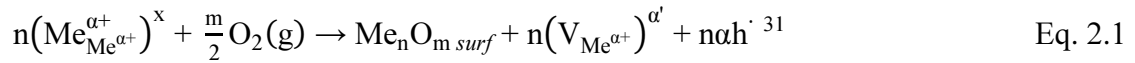


porous while the material near the cathode is Na-rich and has a smooth unbroken surface (**Figure 2.4a** inset images). According to the XRD results, the Na-deficient region is composed of only  $\text{Co}_3\text{O}_4$  crystals while the Na-rich region includes Na-Co-O based stoichiometric phases such as  $\text{Na}_{0.6}\text{CoO}_2$ ,  $\text{Na}_4\text{CoO}_3$  and  $\text{Na}_4\text{CoO}_4$  in addition to  $\text{Co}_3\text{O}_4$  (**Figure A.3**, Appendix A).

Oxygen is a necessary component of the kinetic-demixing to form nanosheets. When the kinetic-demixing is performed under nitrogen gas, the abrupt Na concentration change and the formation of the porous region were diminished to a negligible level compared to when kinetic-demixing is performed under atmospheric conditions. As a comparison, when the kinetic-demixing is performed under nitrogen gas the volume of the Na-deficient region is ~5% of the initial pellet volume, whereas, for the same demixing time and applied current, if the kinetic-demixing is performed under atmospheric conditions the volume of the Na-deficient region is ~45% (**Figure 2.5**).

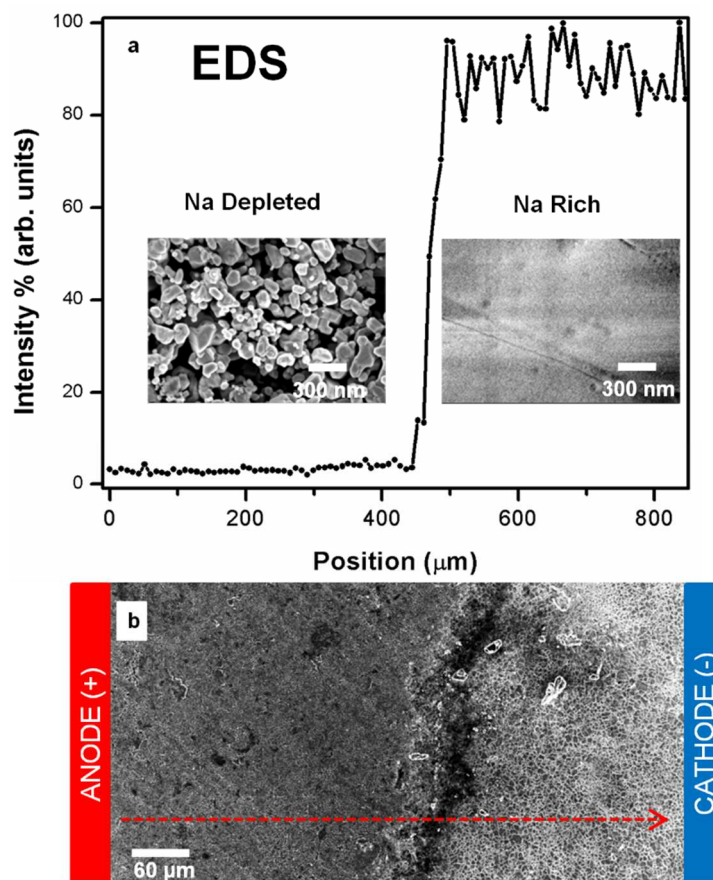
During the E-field induced kinetic-demixing process Co ions are expected to have significantly lower mobility compared to Na ions due to the low temperatures in our process (<300 °C). Extrapolating from previous work on the Co-O system results in extremely low Co tracer diffusion coefficients at temperatures below 300 °C (i.e.,  $D_{\text{Co}}^* = D_{\text{Co}}^{*\theta} \cdot e^{(-E_{\text{Co}}/RT)}$  with  $D_{\text{Co}}^{*\theta} = 1.52 \cdot 10^{-2} \text{ cm}^2/\text{s}$  and  $E_{\text{Co}} = 172400 \text{ J/mol}$ ;  $D_{\text{Co}}^* \approx 3 \cdot 10^{-18} \text{ cm}^2/\text{s}$  at 300 °C).<sup>29</sup> Additionally, Ohta et. al. reported that Co ions were stationary even at higher temperatures (600-700 °C) during reactive solid-phase epitaxial growth of  $\text{Na}_x\text{CoO}_2$  via lateral diffusion of Na into a cobalt oxide film from  $\text{NaHCO}_3$  powder.<sup>30</sup>

The movement of Na and Co ions and the pore formation near the cathode can be best explained by E-field induced kinetic-demixing of multicomponent oxide mixtures. According to Martin's description of E-field induced kinetic-demixing of an initially homogenous multicomponent metal oxide mixture,<sup>31</sup> the cations move towards the cathode due to the electrochemical driving force exerted by the E-field. The cations in such multicomponent oxides, however, don't move as free ions since this usually requires an extremely high amount of energy to liberate the cation from O<sup>2-</sup> (e.g., formation enthalpies of CoO and Co<sub>3</sub>O<sub>4</sub>:  $\Delta_f H_{CoO}^0 = -237.735$  kJ/mol and  $\Delta_f H_{Co_3O_4}^0 = -910.020$  kJ/mol<sup>32</sup>). They instead move by forming new metal oxides at the cathode and by decomposing existing metal oxides at anode. At the cathode, the cations are involved in the chemical reaction described in Eq. 2.1 (for a cation of  $\alpha$ + oxidation state):



In this equation Me represents a metal, V represents vacancy, h represents hole, the *surf* subscript refers to the surface at the electrode and the superscripts  $\alpha'$ ,  $\cdot$ , and x represent  $-\alpha$ , +1 and 0 charges respectively according to the Kröger-Vink notation. This expression conveys the following process: n metal ion(s) in the metal oxide compound (i.e. Me<sub>n</sub>O<sub>m</sub>) reacts with atmospheric oxygen and forms a new metal oxide at the electrode surface (i.e. Me<sub>n</sub>O<sub>m surf</sub>) leaving behind n metal ion vacancy(ies), and n· $\alpha$  free hole(s) are generated to compensate the charge. Here the metal ion can be

either one of the metal cations (Na or Co in our case) in the multicomponent oxide mixture. The same chemical reaction described by Eq. 2.1 is reversed at the anode side.<sup>31</sup> As a result of the reaction described by Eq. 2.1, the oxygen sublattice (formed at the cathode as the new metal oxide such as Na-O, Co-O and/or Na-Co-O stoichiometric compounds) moves towards the anode. The movement of cations, then, is relative to the movement of the oxygen sublattice (analogous to a boat advancing against the flow of a river). The same reaction also generates vacancies at the cathode and terminates them at the anode, which results in a vacancy flow towards anode (see Eq. 2.1). If the mobilities of the two cations are sufficiently different, the high and low mobility cations will separate from this reaction process: the higher mobility cations will fill the newly generated vacancies – due to their high jump frequency – and the lower mobility cations will fill the vacancies emptied by higher mobility cations. As a result, the cation with a higher diffusion coefficient concentrates near the cathode whereas the lower diffusion coefficient cation concentrates near the anode.<sup>31</sup>



**Figure 2.4: The effect of kinetic-demixing.** (a) Na concentration from EDS line scan of the pellet after kinetic-demixing with respective SEM images of the Na depleted and Na-rich regions. A stark contrast is seen between these regions: the Na depleted regions are more porous and contain nano-grains while the surface of the Na-rich region is smooth. (b) SEM image showing cracking between the two regions. The separation line corresponds to the abrupt jump shown in the EDS of (a). Red dashed arrow shows direction of EDS scan.

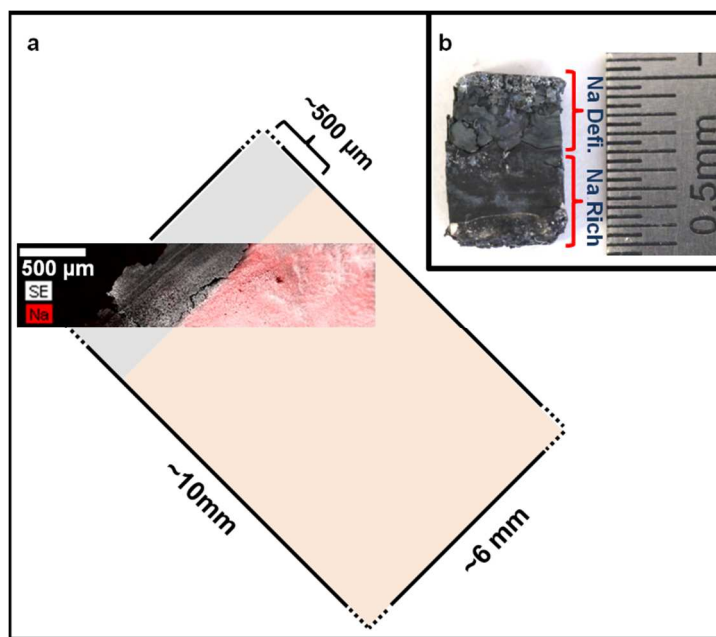
In Martin's description of kinetic-demixing, atmospheric oxygen is necessary for the movement of cations, as is evident in the reaction described by Eq. 2.1.<sup>31</sup> Our

results showing a lack of demixing when performed under nitrogen gas confirm a similar mechanism: Na ions require oxygen to move (demix) through our samples. In this work, the mobility of the Co ions is expected to be much lower than the mobility of Na ions. Therefore newly forming vacancies are filled mostly by Na ions, and different from Martin's observations,<sup>31</sup> in the limited demixing time, the lower mobility Co ions can't fill all of the vacancies created by the movement of Na ions toward the cathode. After the Na ions move from the anode region a porous material develops because Co ions are unable to fill these vacancies.

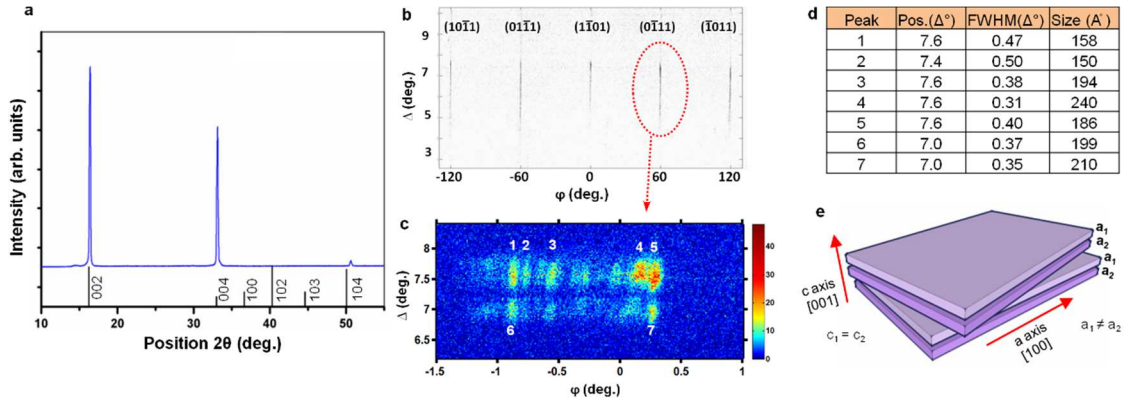
The kinetic-demixing creates an Na saturated compound, which increases the Co diffusion coefficient during the high temperature (1030 °C) calcination, leading to the millimeter-length nanosheets. Correlating the EDS data with the sample volume suggests that the Na:Co ratio can be as high as 1.3 in the Na-rich side which is ~2x higher than the most thermodynamically stable phase of  $\text{Na}_x\text{CoO}_2$ .<sup>33</sup> During the high temperature (1030 °C) calcination, this Na rich part of the pellet shows molten fluidity that is evident from the pellet shape deformation (i.e. rectangular form to semi-spherical form) after the calcination. However, such shape deformation is not observed for the samples calcined directly after pressure application without the kinetic-demixing process (i.e., when the Na:Co ratio is maintained at 0.71). The fluidity in the Na saturated compound at high temperatures is associated with a significant increase in the Co diffusion coefficient. The increased Co mobility is in contrast to their very low mobility in the kinetic-demixing process, which is performed at low temperatures (<300 °C). Assuming the dominant diffusion mechanism is cation vacancy diffusion, this increase in the Co diffusion coefficient

can be explained by the temperature dependence of Co diffusion coefficient and also the “physical correlation effect”<sup>34</sup> between the Co and Na atomic motions. A previous study from Schnehage et al. on the  $\text{Co}_x\text{Mg}_{1-x}\text{O}$  system ( $0 \leq x \leq 1$ ) found the Co tracer diffusion coefficient increases  $\sim 10^9$  times in pure CoO when the temperature increased from 300 °C ( $D_{\text{Co}}^* \approx 3 \cdot 10^{-18} \text{ cm}^2/\text{s}$ ) to 1030 °C ( $D_{\text{Co}}^* \approx 2 \cdot 10^{-9} \text{ cm}^2/\text{s}$ ).<sup>29</sup> According to the same work, the diffusion coefficient of the low mobility Mg atoms can be increased up to  $\sim 500$  times by mixing with the highly mobile Co atoms due to the “physical correlation” between the movements of the two atomic species ( $D_{\text{Mg}}^*$  increases with increasing  $x$ ). Here “physical correlation” refers to a deviation from the directional randomness in atom/vacancy jumps due to the difference in the intrinsic jump frequencies of different atomic species.<sup>34</sup> In the current work the Co ions are the low mobility species and the Na ions are the high mobility species, as is evidenced in the kinetic-demixing process. The diffusion coefficient of the Co cations will therefore be amplified by two factors: 1) the temperature increase and 2) from the “physical correlation” of their mixing with the highly mobile Na ions. Our observation that high temperature molten fluidity exists only kinetically-demixed samples (i.e., if the samples were not kinetically-demixed then there was no molten fluidity) points to increased diffusion from the physical correlation of a larger Na:Co ratio. The Co diffusion amplification significantly reduces kinetic limitations, and leads to long nanosheet formation through cooperative effects including: facile grain growth, fluidity in the pellet during calcination, and enhancement of the anisotropic growth along the (001) plane. The growth along the (001) plane is naturally favored due to the high difference in surface energies between the (001) plane and the planes

perpendicular to (001), which originates from the highly different bond energies along corresponding directions.<sup>35</sup>



**Figure 2.5: Na-deficient and Na-rich regions after E-field induced kinetic-demixing performed under nitrogen gas (a) and atmospheric conditions (b).** (a) An SEM image with secondary-electron EDS mapping for Na – shown in red – for a sample pellet demixed under nitrogen gas. The black outer lines mark the approximate boundaries of the full pellet. The SEM rectangle shows the false-color mapping for Na in red, and the SEM image position relative to the full pellet. The Na-deficient region is only ~5% of the total volume, indicating that very little kinetic-demixing occurred. (b) Optical image of the pellet which is demixed under atmospheric conditions. Under these conditions 45% of the pellet is now Na-deficient.



**Figure 2.6: X-ray Characterization of Nanosheets.** (a) Conventional  $\theta$ -2 $\theta$  XRD of Na<sub>0.7</sub>CoO<sub>2</sub> nanosheets aligned parallel to sample stage (stick lines correspond to pdf card file #00-030-1182). (b) Grazing incidence diffraction from hexagonal {10 $\bar{1}$ 0} planes of Na<sub>0.7</sub>CoO<sub>2</sub> nanosheets ( $\phi$  scan with linear detector around 101 peak) indicates 6-fold symmetry with peaks around 7.5° ( $\Delta$ ) located 60° ( $\phi$ ) away from each other. (c) Higher resolution of the region in (b) with the seven most distinct diffraction spots numbered. (d) Finite sizes calculated using Scherrer's equation for the seven spots in (c). (e) 3D Sketch of a hypothetical ordering of nanosheets according to available diffraction data.

Anisotropic grain growth behavior of kinetically-demixed samples was analyzed comparing them with non-kinetically-demixed samples at several calcination temperatures (650 to 1030 °C, **Figure A.4**, Appendix A). For both kinetically-demixed and non-kinetically-demixed samples the nanosheet *thickness* does not change with calcination temperature, but the nanosheet *length* increases with increasing calcination temperature. The nanosheet lengths are similar for the kinetically demixed and non-kinetically-demixed samples at 650 °C, 750 °C and 850

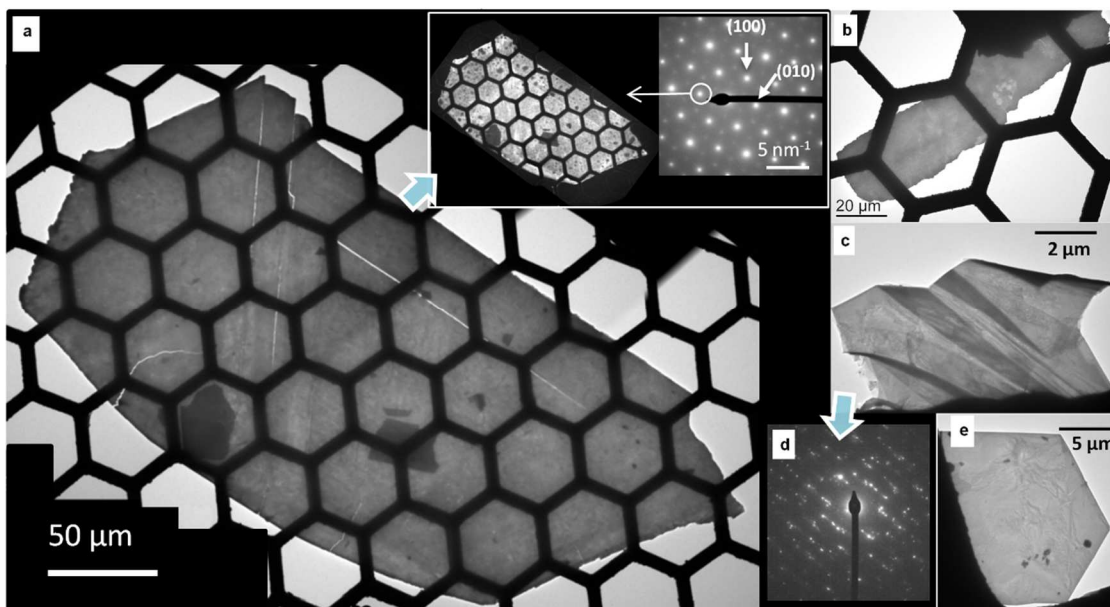


°C (i.e., ~1  $\mu\text{m}$ , ~5  $\mu\text{m}$  and ~13  $\mu\text{m}$  respectively). However, at temperatures higher than 950 °C the nanosheets for the kinetically-demixed samples become significantly longer compared to non-kinetically-demixed samples. Nanosheets resulting from kinetic-demixed samples with calcination at 950 °C (the fluidity of the pellet is not observed at this temperature) are ~4 times longer than the non-kinetically-demixed samples. At 1030 °C, the nanosheets resulting from the kinetically-demixed samples are ~9 times longer than the nanosheets resulting from the non-kinetically-demixed samples. The nanosheet length can be as long as 1.8 mm for the kinetically-demixed samples calcined at 1030 °C.

The  $\theta$ -2 $\theta$  XRD study of the crushed powders after calcination indicates trigonal  $\text{Na}_{0.9}\text{CoO}_2$ <sup>19</sup> as the dominant phase. This phase can be transformed to the hexagonal  $\text{Na}_{0.7}\text{CoO}_2$ <sup>18</sup> phase after a secondary heat treatment (875 °C-30 hr) (**Figure A.3**, Appendix A).<sup>36</sup> The transition from  $x=0.9$  to 0.7 has no effect on nanosheet thickness and length.  $\theta$ -2 $\theta$  XRD scans show only the 00L and 104 peaks for the millimeter-long nanosheets aligned parallel to the sample stage, while other diffraction peaks are absent (**Figure 2.6a**). This result indicates that the nanosheets are large crystals which are strongly oriented.

Conventional  $\theta$ -2 $\theta$  XRD scans of the 2D crystals show no finite size broadening of the 00L peaks (**Figure 2.6a**). The 6-angle diffractometer based XRD analysis conducted at the CHESS provides a more accurate description of the 2D crystal crystals and finite-size effects (see **Figure A.5**, Appendix A). Results from this XRD analysis show that the crystal symmetry is consistent with  $\text{Na}_{0.7}\text{CoO}_2$ . Peaks at ~7.5° from the horizontal ( $\Delta$ ) and separated by 60° along the in-plane  $\phi$  axis

correspond to the six-fold symmetry of the (101) planes of  $\text{Na}_{0.7}\text{CoO}_2$  (**Figure 2.6b**). Higher resolution scans of these peaks from several samples show  $\sim 4\text{--}12$  diffraction spots within  $\sim 1^\circ$  along the  $\phi$  axis. **Figure 2.6c** shows the case with  $\sim 12$  diffraction spots. Because the  $\phi$  scan represents a rocking curve for (100) planes, these spots are the result of rotational misalignment ( $\sim 0.1^\circ$  with respect to each other) of the  $\sim 12$  distinct crystals around [001].



**Figure 2.7: TEM images of nanosheets of  $\text{Na}_{0.7}\text{CoO}_2$ .** (a)  $350\text{ }\mu\text{m} \times 150\text{ }\mu\text{m} \times 100\text{ nm}$  thick sheet composed of  $\sim 5$  individual nanosheets. Image was compiled from 3 electron micrographs. Inset: ED pattern confirming nanosheets are aligned, single crystal and the orientation of cross-plane thickness is [001]. Dark-field imaging confirms that this ED pattern is representative of the entire sheet and that the sheet is a single crystal. (b)  $120\text{ }\mu\text{m} \times 25\text{ }\mu\text{m} \times 100\text{ nm}$  thick nanosheet as another example of large scale exfoliation (c) Image of folded sheet and (d) corresponding ED pattern. (e) Micron sized sheet with hexagonal growth morphology matching unit cell.

The CHES diffraction results also show peak broadening associated with the 2D crystal thickness (**Figure 2.6b** and **2.6c**). Finite-size broadening effects can be analyzed by examining the (101) diffraction. Since the  $\Delta$  and  $\phi$  axes are orthogonal in both real and reciprocal space, and  $\Delta$  axis projects onto the vertical linear detector which is parallel to [001], vertical broadening of the 101 peak corresponds to finite size effects along the [001] direction (see **Figure A.6c**, Appendix A). According to Scherrer analysis (**Figure 2.6d**), the broadening along the  $\Delta$  axis gives an average nanosheet thickness of 19.1 nm, which is in good agreement with statistical analysis performed by SEM (18.2 nm, **Figure A.2**, Appendix A). It is surprising that finite size broadening is only observed in the high-resolution 2D scan, but not in conventional  $\theta$ -2 $\theta$  XRD of (001) planes. The likely explanation why broadening is not observed for the conventional  $\theta$ -2 $\theta$  XRD is because the nanosheets are stacked in registry along [001]. Broadening is observed in the in-plane 2D scans of the (101) peak because these planes are not contiguous between nanosheets, and thus we are able to tease out the finite size of the nanosheets along [001]. Such stacking of nanosheets is referred to as “turbostratic” arrangement of the 2D crystals (planes rotated around the  $c$ -axis, see **Figure A.6b**, Appendix A).<sup>37</sup> Presence of this arrangement is substantiated by the rotational misalignment of the  $\sim 12$  grains (**Figure 2.6c**).

The two distinct diffraction spots observed along  $\Delta$  axis indicate two different (100) lattice spacings. A hypothetical 3D sketch of the possible 2D crystal stacking configuration is provided in **Figure 2.6e**. It is important to note that no broadening

was observed for either of the two (100) d-spacings. This observation confirms that the finite size along [001] solely originates from the nanosheet thickness.

Extremely large  $\text{Na}_{0.7}\text{CoO}_2$  nanosheets are exfoliated from the bulk, stacked nanosheet samples and their crystal properties are characterized. Free-standing exfoliated 2D crystals are typically 10–350 microns wide, as characterized using optical microscopy and TEM (**Figure 2.7** and **Figure A.7**, Appendix A). This length (350 microns) yields an order of magnitude improvement in the maximum lateral lengths of typical metal oxide 2D crystals (nanosheets).<sup>4</sup> The exfoliated pieces are estimated to be between 20–100 nm thick based on layer counting at the sample edges (**Figure A.8**, Appendix A). SAED from multiple regions of each 2D crystal confirm the hexagonal crystal symmetry and lattice spacing of  $\text{Na}_{0.7}\text{CoO}_2$  with [001] being the zone axis. The well-defined points in SAED (**Figure 2.7a**) confirm that the stacked nanosheets are in registry with one another along the c-direction. TEM dark-field analysis from a hexagonal-lattice {100} diffraction spot shows the entire sheet illuminated for the exfoliated nanosheets, proving that the entire sheet is a single crystal (**Figure 2.7a** inset). In reflected white light (**Figure A.7a**, Appendix A) the nanosheets appear opaque and black, consistent with bulk samples of  $\text{Na}_{0.7}\text{CoO}_2$ . However, under transmitted light they range from a translucent yellow to red (**Figure A.7b**, Appendix A). This finding yields a facile method to distinguish between thin, electron transparent layers and bulk pieces of  $\text{Na}_{0.7}\text{CoO}_2$ . Cross-polarized transmitted light shows the sheets to be optically isotropic. This observation is expected when viewing a hexagonal crystal along [001].

## 2.4. Conclusions

A novel synthesis has been developed for production of millimeter-length nanosheets of  $\text{Na}_x\text{CoO}_2$ . Our nanomanufacturing method is scalable and low-cost, capable of producing tens of thousands of nanosheets for device integration. The nanosheets have thicknesses in the tens of nanometers while their lateral lengths are millimeters, resulting in a very high anisotropic aspect ratio (10-5:1:1). Synchrotron x-ray studies indicated that the nanosheets are turbostratically stacked within the pellets. The nanosheets are readily exfoliated into free-standing sheets reaching 350 microns in length with thicknesses ranging ~20–100 nm. SAED studies indicated that the crystal properties are maintained after the exfoliation process. This new nanomanufacturing method can be applicable to other atomically layered oxides. (See **Figure A.9, Appendix A** for preliminary thermoelectric measurements of  $\text{Na}_{0.7}\text{CoO}_2$  and  $\text{Na}_{0.9}\text{CoO}_2$  nanosheet stacks.)

## REFERENCES

- <sup>1</sup> M. Aksit, D. P. Toledo, and R. D. Robinson, *J. Mater. Chem.* **22** (13), 5936 (2012).
- <sup>2</sup> M. Osada and T. Sasaki, *J. Mater. Chem.* **19** (17), 2503 (2009).
- <sup>3</sup> J. N. Coleman, M. Lotya, A. O'Neill, S. D. Bergin, P. J. King, U. Khan, K. Young, A. Gaucher, S. De, R. J. Smith, I. V. Shvets, S. K. Arora, G. Stanton, H. Y. Kim, K. Lee, G. T. Kim, G. S. Duesberg, T. Hallam, J. J. Boland, J. J. Wang, J. F. Donegan, J. C. Grunlan, G. Moriarty, A. Shmeliov, R. J. Nicholls, J. M. Perkins, E. M. Grieveson, K. Theuwissen, D. W. McComb, P. D. Nellist, and V. Nicolosi, *Science* **331** (6017), 568 (2011).
- <sup>4</sup> R. Ma and T. Sasaki, *Adv. Mater.* **22** (45), 5082 (2010).
- <sup>5</sup> A. Takagaki, M. Sugisawa, D. Lu, J. N. Kondo, M. Hara, K. Domen, and S. Hayashi, *J Am Chem Soc* **125** (18), 5479 (2003).
- <sup>6</sup> L. Z. Wang, K. Takada, A. Kajiyama, M. Onoda, Y. Michiue, L. Q. Zhang, M. Watanabe, and T. Sasaki, *Chem Mater* **15** (23), 4508 (2003).
- <sup>7</sup> L. Zhang, X. Tang, and W. Gao, *Cryst. Growth Des.* **8** (7), 2489 (2008).
- <sup>8</sup> W. Sugimoto, H. Iwata, Y. Yasunaga, Y. Murakami, and Y. Takasu, *Angewandte Chemie-International Edition* **42** (34), 4092 (2003).
- <sup>9</sup> S. Suzuki and M. Miyayama, *J. Phys. Chem. B* **110** (10), 4731 (2006).
- <sup>10</sup> S. Suzuki and M. Miyayama, *J Electrochem Soc* **154** (5), A438 (2007).
- <sup>11</sup> H. Sato, K. Okamoto, K. Tamura, H. Yamada, K. Saruwatari, T. Kogure, and A. Yamagishi, *Applied Physics Express* **1** (3), 35001 (2008).

- <sup>12</sup> M. Osada, Y. Ebina, K. Takada, and T. Sasaki, *Adv. Mater.* **18** (3), 295 (2006).
- <sup>13</sup> K. Koumoto, Y. Wang, R. Zhang, A. Kosuga, and R. Funahashi, *Annual Review of Materials Research* **40** (1), 363 (2010).
- <sup>14</sup> Y. Wang, Y. Sui, P. Ren, L. Wang, X. J. Wang, W. H. Su, and H. J. Fan, *Inorg. Chem.* **49** (7), 3216 (2010).
- <sup>15</sup> W. Liu and M. Asheghi, *J. Appl. Phys.* **98** (12), 123523 (2005).
- <sup>16</sup> I. Terasaki, Y. Sasago, and K. Uchinokura, *Phys. Rev. B* **56** (20), 12685 (1997).
- <sup>17</sup> Y. Y. Wang, N. S. Rogado, R. J. Cava, and N. P. Ong, *Nature* **423** (6938), 425 (2003).
- <sup>18</sup> Q. Huang, M. L. Foo, R. A. Pascal, J. W. Lynn, B. H. Toby, T. He, H. W. Zandbergen, and R. J. Cava, *Phys. Rev. B* **70** (18), 184110 (2004).
- <sup>19</sup> L. Viciu, J. W. G. Bos, H. W. Zandbergen, Q. Huang, M. L. Foo, S. Ishiwata, A. P. Ramirez, M. Lee, N. P. Ong, and R. J. Cava, *Phys. Rev. B* **73** (17), 174104 (2006).
- <sup>20</sup> I. Terasaki, I. Tsukada, and Y. Iguchi, *Phys. Rev. B* **65** (19), 195106 (2002).
- <sup>21</sup> Y. Masuda, Y. Hamada, W. S. Seo, and K. Koumoto, *J. Nanosci. Nanotechnol.* **6** (6), 1632 (2006).
- <sup>22</sup> D. M. Smilgies, D. R. Blasini, S. Hotta, and H. Yanagi, *Journal of Synchrotron Radiation* **12**, 807 (2005).
- <sup>23</sup> M. P. Pechini, U.S. Patent No. 3,330,697 (1967).
- <sup>24</sup> S. Sakka, *Handbook of sol-gel science and technology: processing,*

*characterization and applications*. (Kluwer Academic Publishers, 2005).

- <sup>25</sup> M. A. Gulgun, M. H. Nguyen, and W. M. Kriven, *J Am Ceram Soc* **82** (3),  
556 (1999).
- <sup>26</sup> M. Kakihana, *J. Sol-Gel Sci. Technol.* **6** (1), 7 (1996).
- <sup>27</sup> D. A. Rees and E. J. Welsh, *Angewandte Chemie-International Edition* **16** (4),  
214 (1977).
- <sup>28</sup> L. Zhang, X. F. Tang, and W. B. Gao, *Journal of Electronic Materials* **38** (7),  
1229 (2009).
- <sup>29</sup> M. Schnehage, R. Dieckmann, and H. Schmalzried, *Berichte Der Bunsen-  
Gesellschaft-Physical Chemistry Chemical Physics* **86** (11), 1061 (1982).
- <sup>30</sup> H. Ohta, S. W. Kim, S. Ohta, K. Koumoto, M. Hirano, and H. Hosono, *Cryst.  
Growth Des.* **5** (1), 25 (2005).
- <sup>31</sup> M. Martin, *Solid State Ionics* **136**, 331 (2000).
- <sup>32</sup> M.W. Chase, *JANAF thermochemical tables*. (American Chemical Society,  
1986).
- <sup>33</sup> Y. S. Meng, Y. Hinuma, and G. Ceder, *J Chem Phys* **128** (10), 104708 (2008).
- <sup>34</sup> J. Philibert, *Atom movements: diffusion and mass transport in solids*. (Éditions  
de Physique, 1991).
- <sup>35</sup> C. T. Lin, D. P. Chen, P. Lemmens, X. N. Zhang, A. Maljuk, and P. X. Zhang,  
*Journal of Crystal Growth* **275** (3-4), 606 (2005).
- <sup>36</sup> L. W. Shacklette, T. R. Jow, and L. Townsend, *J Electrochem Soc* **135** (11),  
2669 (1988).
- <sup>37</sup> K. Ufer, G. Roth, R. Kleeberg, H. Stanjek, R. Dohrmann, and J. Bergmann,



Zeitschrift Fur Kristallographie **219** (9), 519 (2004).

## CHAPTER 3

### 3. SYNTHESIS AND PROPERTIES OF ELECTRICALLY CONDUCTIVE, DUCTILE, EXTREMELY LONG ( $\sim 50\ \mu\text{m}$ ) NANOSHEETS OF $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$ \*

#### 3.0. *Abstract*

Extremely long, electrically conductive, ductile, free-standing nanosheets of water-stabilized  $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$  are synthesized using the SGKD process. Room temperature in-plane resistivity of the  $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$  nanosheets is less than  $\sim 4.7\ \text{m}\Omega\cdot\text{cm}$ , which corresponds to one of the lowest resistivity values reported for metal oxide nanosheets. The synthesis produces tens of thousands of very high aspect ratio ( $50,000:50,000:1$  - length:width:thickness), millimeter length nanosheets stacked into a macro-scale pellet. Free-standing nanosheets up to  $\sim 50\ \mu\text{m}$  long are readily delaminated from the stacked nanosheets. HR-TEM studies of the free-standing nanosheets indicate that the delaminated pieces consist of individual nanosheet crystals that are turbostratically stacked. XRD studies confirm that the nanosheets are stacked in perfect registry along their c-axis. SEM based statistical analysis show that the average thickness of the nanosheets is  $\sim 13\ \text{nm}$ . The nanosheets show ductility with a bending radius as small as  $\sim 5\ \text{nm}$ .<sup>1</sup>

#### 3.1. *Introduction*

Ultrathin metal oxide sheets can exhibit quantum size and surface effects that

---

\* Originally Published as: Mahmut Aksit, Ben C. Hoselton, Ha Jun Kim, Don-Hyung Ha and Richard D. Robinson, "Synthesis and Properties of Electrically Conductive, Ductile, Extremely Long ( $\sim 50\ \mu\text{m}$ ) Nanosheets of  $\text{K}_x\text{CoO}_2\cdot y\text{H}_2\text{O}$ ", *ACS Applied Materials & Interfaces* 5 (18), 8998-9007 (2013). Reprinted with Permission from American Chemical Society.

result in novel electronic, ferromagnetic, magneto-optical, electrochemical, catalytic, and photoresponsive properties.<sup>2-11</sup> However, large scale synthesis of inorganic, free-standing two dimensional (2D) nanosheet materials still remains a challenge, especially with compounds such as transition metal oxides.<sup>4-6,12</sup>

Free-standing nanosheets of transition metal oxides have exciting technological implications and are promising candidates for replacing conventional thin films in certain applications. 2D metal oxide nanosheets can provide extremely large surface areas with high chemical and mechanical durability, leading to efficient charge storage. For instance, 1) lithium-ion battery cathodes of nanoporous  $\text{LiMn}_2\text{O}_4$  nanosheets display superior cycling performance compared to bulk  $\text{LiMn}_2\text{O}_4$  at high charge-discharge rates,<sup>13</sup> 2) octatitanate nanosheets have better reversible capacities than those of conventional octatitanate, presumably due to the ability of the nanosheets to better withstand damage caused by lithium insertion,<sup>14,15</sup> and 3) highly efficient and stable pseudocapacitive cathodes have been obtained from  $\text{K}_{0.15}\text{MnO}_2 \cdot 0.43\text{H}_2\text{O}$  nanosheets.<sup>16</sup> Especially for nano-scale high- $\kappa$  dielectrics, free-standing metal oxide nanosheets perform better than metal oxide thin films grown by conventional vapor deposition techniques, which tend to show decreasing dielectric constants with decreasing film thickness.<sup>17,18</sup> For example, < 10 nm thick nanosheets of  $\text{Ti}_{0.87}\text{O}_2$ ,  $\text{Ca}_2\text{Nb}_3\text{O}_{10}$  and  $\text{Sr}_2\text{Nb}_3\text{O}_{10}$  used as gate dielectric layers provided high dielectric constants ( $\epsilon_r \approx 125$ , 210 and 240 respectively) and very low leakage current densities in FET devices.<sup>4,19</sup> Metal oxide nanosheets can also be utilized as semiconducting materials in microelectronic devices as previously demonstrated with FET semiconducting channels made from  $\text{Ti}_{0.91}\text{O}_2$  nanosheets.<sup>4</sup> Free-standing metal oxide

nanosheets have the advantage of being solution processable, allowing deposition on substrates for device integration. Because they don't need to be grown on a substrate, free-standing nanosheets have different boundary and surface conditions from conventional thin films, which can result in useful physical properties.

Producing free-standing 2D nanostructures of alkali cobaltates is difficult to achieve by the conventional method for nanosheet synthesis – chemical exfoliation – because the chemical treatment usually causes complete depletion of alkali content from the crystal structure, resulting in CoO nanosheets.<sup>4,6,20</sup> In our recent study, we reported millimeter-length, high aspect ratio nanosheets of  $\text{Na}_x\text{CoO}_2$ , which could be readily delaminated into free-standing nanosheets without changing crystal structure.<sup>21</sup> As an atomically layered complex metal oxide, bulk  $\text{Na}_x\text{CoO}_2$  has fascinating stoichiometric-dependent properties such as a high thermoelectric power factor and high electrical conductivity.<sup>22-27</sup> Nanosheets of  $\text{Na}_x\text{CoO}_2$  could result in improved thermoelectric performance compared to bulk because of phonon scattering and confinement effects,<sup>28</sup> and they could be used as conductive supports for electrocatalytic applications due to their high surface area and high oxidation resistance.<sup>29</sup> Nanosheets of  $\text{K}_x\text{CoO}_2$  should also be important for practical applications because of the similarities between  $\text{K}_x\text{CoO}_2$  and  $\text{Na}_x\text{CoO}_2$  in terms of the crystal family (hexagonal), host atomic layer ( $\text{CoO}_2$  octahedra) and intercalating atomic species (alkali metals).<sup>30,31</sup>

$\text{K}_x\text{CoO}_2$  has a similar crystal structure<sup>31</sup> to  $\text{Na}_x\text{CoO}_2$  and has also been researched for its desirable electronic transport properties. The number of studies of  $\text{K}_x\text{CoO}_2$ , however, is limited, likely due to the extreme hygroscopicity of  $\text{K}_x\text{CoO}_2$ .<sup>31-34</sup>

$K_xCoO_2$  consists of two dimensional triangular  $CoO_2$  layers with  $K^+$  ion intercalated between the layers.<sup>31</sup> The K:Co ratio and the sites occupied by  $K^+$  ions in the crystal vary with the different phases (x) of  $K_xCoO_2$ .<sup>34</sup> At room temperature, the charge-ordered hexagonal  $K_{4/7}CoO_2$  phase has the lowest electrical resistivity ( $\sim 10 \text{ m}\Omega\cdot\text{cm}$ ) among the phases of  $K_xCoO_2$ ,<sup>34</sup> whereas the largest Seebeck coefficient (thermopower) was observed for the rhombohedral  $K_{0.5}CoO_2$  phase ( $\sim 30 \text{ }\mu\text{V/K}$ ).<sup>31</sup> Tang *et al.* reported that the water stabilized  $K_{0.35}CoO_2\cdot 0.34H_2O$  phase has  $\sim 4$  times lower electrical resistivity ( $\sim 2.3 \text{ m}\Omega\cdot\text{cm}$ ) compared to the  $K_{4/7}CoO_2$  phase.<sup>35</sup> Besides having lower electrical resistivity,  $K_{0.35}CoO_2\cdot 0.34H_2O$  is also more stable under atmospheric conditions compared to the anhydrous  $K_xCoO_2$  phases. The high electrical conductivity and atmospheric stability of  $K_{0.35}CoO_2\cdot 0.34H_2O$  make it a promising candidate for practical applications.

Extremely long ( $>50$  microns), bendable nanosheets of electrically conductive  $K_xCoO_2\cdot yH_2O$  are of particular interest because they can be utilized in metal oxide nanoelectronics as electrical conductors.<sup>4</sup> Although many metal oxide nanosheets have been reported in the literature for use in oxide nanoelectronics, the majority of these nanosheets are either semiconductors or insulators with dielectric properties.<sup>4,36</sup> The number of studies on high electrical conductivity metal oxide nanosheets has been limited.<sup>37,38</sup> Among the metal oxide nanosheets in the literature,  $RuO_2$  nanosheets are the only room temperature electrical conductors with a low sheet resistance ( $R_s$ ), at  $12 \text{ k}\Omega \text{ sq}^{-1}$  for a  $1.38 \text{ nm}$  thick ( $t$ ) single layer (corresponding to a resistivity value of  $\rho = R_s\cdot t \approx 1.7 \text{ m}\Omega\cdot\text{cm}$ ) and  $0.36 \text{ k}\Omega \text{ sq}^{-1}$  for a ten-layered film ( $\rho = R_s\cdot t \approx 0.5 \text{ m}\Omega\cdot\text{cm}$ , assuming  $t \approx 13.8 \text{ nm}$ ).<sup>37</sup> However, the lateral sizes of these nanosheets are smaller

than several micrometers, and making them impractical for large-scale electronic devices. Here we report  $K_xCoO_2 \cdot yH_2O$  nanosheets that could readily serve as long and flexible electrical conductors for metal oxide nanoelectronics due to their large lateral lengths (micron to millimeter scales), relatively high electrical conductivity, and extreme ductility.

In this work, scalable nanomanufacturing of electrically conductive  $K_xCoO_2 \cdot yH_2O$  nanosheets is reported for the first time. The synthesis method is based on the new SGKD process.<sup>21</sup> The final product consists of tens of thousands of well-defined nanosheets that are stacked into a macro-sized pellet. Such a large-scale, bottom-up nanocrystal growth technique is more efficient than conventional nanofabrication and crystal growth techniques. The nanosheets have very high aspect ratios of nanometer thickness and millimeter lateral lengths (50,000: 50,000:1). Delaminated nanosheets are up to ~50 microns long. The length and shape of the nanosheets are uniform. The stacks of the  $K_xCoO_2 \cdot yH_2O$  nanosheets can be readily delaminated into free-standing nanosheets without altering crystal structure. The  $K_xCoO_2 \cdot yH_2O$  nanosheets show extreme ductility when mechanically bent. Electrical resistivity measurements of the  $K_xCoO_2 \cdot yH_2O$  nanosheet stacks indicate the resistivity is lower than ~4.7 m $\Omega$ ·cm at room temperature.

### **3.2. Experimental**

#### **3.2.0. Synthesis of $K_xCoO_2 \cdot yH_2O$ nanosheets**

Appropriate quantities of PAA (average molecular weight:  $M_w \sim 1800$ ), cobalt(II) nitrate hexahydrate (0.230 M) and potassium carbonate (0.115 M) were dissolved in de-ionized water at room temperature. The K to Co ratio is set to 1. The

ratio of PAA carboxylate groups to total metal ions is 1:2. The solution is evaporated at 150 °C on a hotplate with continuous stirring until it reached 20% of the initial volume. The resulting viscous, dark red solution is then autocombusted at 500 °C. The resulting black powder of homogenously mixed K and Co oxides is uni-axially pressed into pellets with a rectangular die set at 400 MPa. A typical size of the pellet is 10×6×5 mm (L×W×t). The pellet is kinetically-demixed for 72 hours at ~400 °C with a constant electrical current of 500 mA passing through Cu plates and contacts made of silver epoxy. Over the course of 72 hours the voltage fluctuates between 20 V and 5 V, and decreases by time. After the kinetic-demixing, the pellet separates into K-rich and K-depleted regions and the boundary between the two regions is clearly distinguishable. The K-depleted region is mechanically weak, porous and grey while the K-rich region is mechanically strong, dense and black. The K-rich region of the pellet is separated and calcined in a tube furnace to obtain  $K_xCoO_2$  nanosheets. The furnace is rapidly heated to 1000 °C and held for 30 minutes. The temperature is then ramped down to 970 °C in 1.5 hours and after reaching 970 °C the furnace is allowed to cool down. The sample is removed from the furnace at around ~700 °C and quickly evacuated in a vacuum desiccator to minimize  $K_2CO_3 \cdot 1.5H_2O$  contamination. During the calcination the K-rich pellet is suspended on 0.25 diameter Au wires to limit the K and Co diffusion to the surrounding environment.  $K_xCoO_2$  nanosheets form as stacks within the calcined pellet. The calcined pellet (nanosheet composite, consisting of large number of  $K_xCoO_2$  nanosheet stacks) was immersed in liquid nitrogen to mechanically extract individual stacks of  $K_xCoO_2$  nanosheets with >20 μm thickness (containing >1500 single nanosheets) and < 2.1 mm length. The nanosheet stacks are then soaked in clean (18 MΩ), room temperature water (250 ml), and stirred using a magnetic stirrer for 4-5 days in order to obtain  $K_xCoO_2 \cdot yH_2O$  nanosheets. Unlike  $K_xCoO_2$  nanosheets, water stabilized  $K_xCoO_2 \cdot yH_2O$  nanosheets are stable enough in

atmospheric conditions to perform XRD, electron microcopy and electrical resistivity measurements without formation of unwanted surface contamination during the measurement process.

### **3.2.1. Delamination of the nanosheets**

Delamination of the  $K_xCoO_2 \cdot yH_2O$  nanosheets is accomplished through ball milling. Stacks of  $K_xCoO_2 \cdot yH_2O$  nanosheets are ball milled in clean, room temperature water for 4-5 hours with 5 mm diameter high-wear-resistant zirconia balls. The delaminated nanosheets are then filtered through a 600 mesh Cu TEM grid (carbon-free) and imaged on the same grid.

### **3.2.2. XRD, SEM, EDS, WDS and TEM characterization**

All of the intermediate and final products of the synthesis procedure were examined by a high resolution ( $\sim 1$  nm at 2 keV) SEM (LEO 1550 FESEM) and by XRD. Conventional  $2\theta$ - $\theta$  XRD measurements were conducted using Rigaku Smartlab and Scintag (Pad-X) instruments and area detector XRD measurements was performed using Bruker General Area Detector Diffraction System. Cu  $K\alpha 1$  x-ray sources were used in all XRD characterization. In order to observe the texturing of the stacks of  $K_xCoO_2 \cdot yH_2O$  nanosheets, the nanosheet stacks were aligned perpendicular to x-ray scattering axis for the XRD measurements. For the XRD-based phase identification, the samples were ground before the XRD measurements in order to minimize possible texturing and crystal alignment effects which can alter relative intensities of different XRD peaks. For the XRD characterization of hygroscopic  $K_xCoO_2$  nanosheets, the samples were first heated up to 500 °C in order to decompose the existing



$\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination and then allowed to cool down to room temperature under  $^4\text{He}$  gas flow. XRD measurements of  $\text{K}_x\text{CoO}_2$  nanosheets were performed at room temperature under continuous  $^4\text{He}$  gas flow to prevent formation of  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination during the measurement. TEM, dark field imaging, and ED were performed with an FEI T12 Spirit TEM at 120 kV accelerating voltage. HR-TEM images were taken with an FEI F20 TEM at 200 kV accelerating voltage. Elemental analyses were performed using an EDS attached to a LEO 1550 FESEM and WDS attached to a JEOL 8900 Microprobe.  $\text{KAlSi}_3\text{O}_8$  and metallic Co are used as reference materials in WDS characterization for high precision quantification of K:Co atomic ratio in  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets.

### 3.2.3. *Electrical Resistivity Measurements*

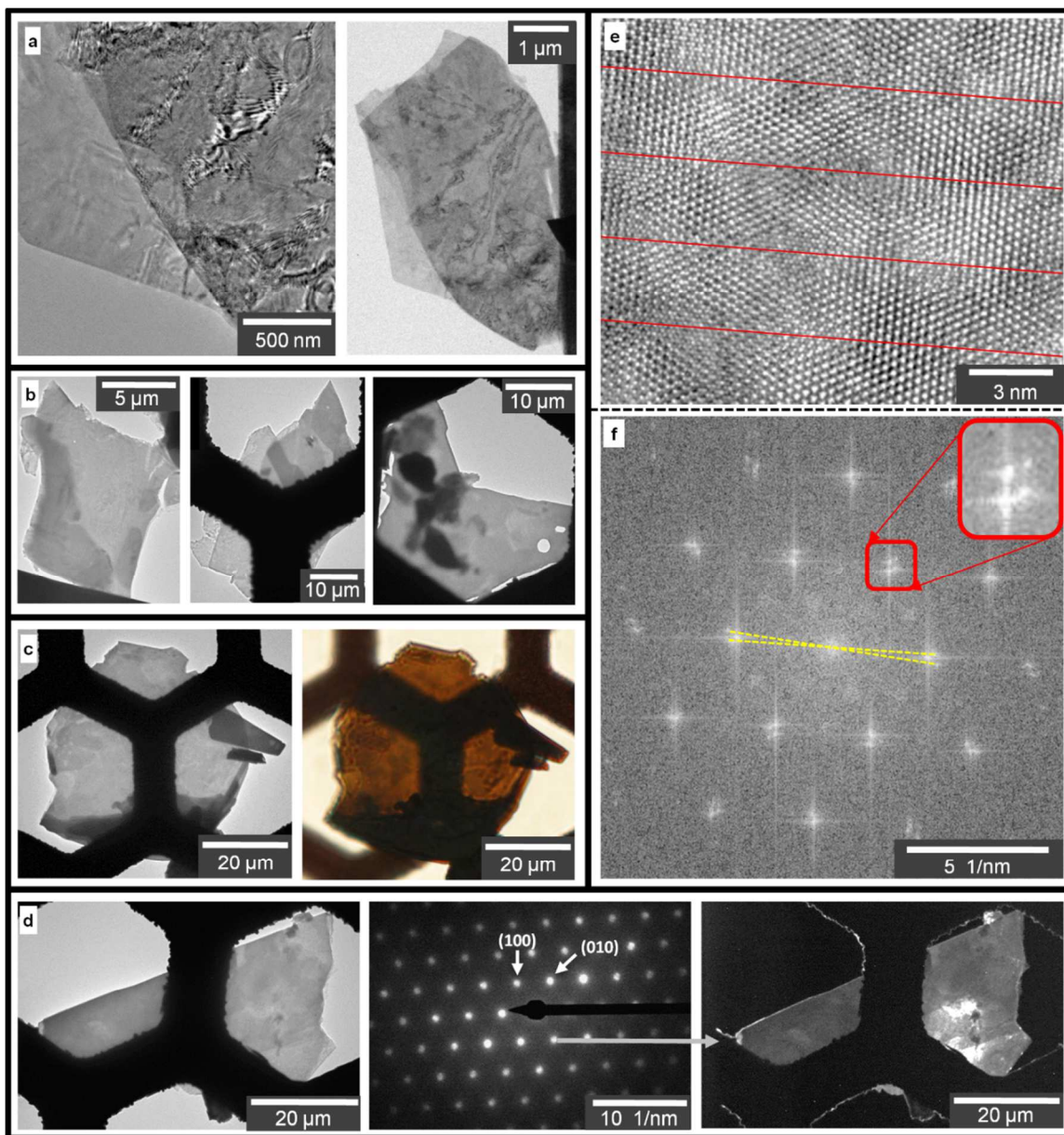
Successful two-point resistivity measurements have been achieved by first cutting thin strips of double sided Scotch Tape to widths comparable to the length of the nanosheet sample. This strip is adhered to a glass slide, and the nanosheet sample is placed on the strip of tape such that two opposing edges protrude past the tape on either side. Conductive silver paint is used to attach the nanosheet samples to four copper electrodes – two electrodes per edge (See **Figure B.1**, Appendix B). The electrodes are short lengths (~2 cm) of 0.08 mm diameter copper wire. The wires are first dipped in soldering flux to remove any native oxide on the copper metal. The silver paint is allowed to dry, and the sample is then connected for a 4-wire resistance measurement by using the four copper electrodes for the voltage and current contacts (See **Figure 3.2** inset drawing). Once attached, the sample is placed in vacuum at <

10 microbar, submerged in a liquid nitrogen fridge, and slowly cooled to 160 K. A resistance measurement is taken every 5 K during cool down.

### 3.3. Results and Discussion

The free-standing nanosheets were characterized by SEM, EDS, TEM and HR-TEM. Very large  $K_xCoO_2 \cdot yH_2O$  nanosheets are delaminated from pellets composed of millimeter-length stacked nanosheets (see methods). The free-standing nanosheets are typically 5–50 microns laterally, as determined by optical microscopy and TEM (**Figure 3.1a**, **3.1b**, **3.1c** and **3.1d**). Thin individual and stacked nanosheets are seen in **Figure 3.1a**. Typical nanosheet stacks with tens-of-microns lateral lengths are shown in **Figure 3.1b**. Under transmitted white light the delaminated nanosheets are translucent brown (**Figure 3.1c** (right)) as previously observed for  $Na_xCoO_2$  nanosheets.<sup>21</sup> This delaminated nanosheet stack in **Figure 3.1c** is ~170 nm thick based on SEM characterization (See **Figure B.2**, Appendix B), and contains ~10 individual nanosheets. SAED from multiple regions of each 2D crystal confirm the hexagonal crystal symmetry with [001] being the zone axis. The well-defined, bright points in SAED (**Figure 3.1d** (middle)) indicate that the stacked nanosheet crystals are in registry with each other along the c-axis direction. TEM dark-field imaging from a hexagonal-lattice {100} diffraction spot shows that the entire sheet is single crystalline with the lateral length being along the <100> directions (**Figure 3.1d** (right)). The lattice spacing and hexagonal symmetry resulting from HR-TEM characterization of a nanosheet (**Figure 3.1e**) and its reciprocal lattice image derived through the Fast Fourier transform (FFT) (**Figure 3.1f**), indicates the nanosheets are crystalline for

$K_x\text{CoO}_2$ . Inspection of the reciprocal lattice points (**Figure 3.1f**, inset) indicate four different nanosheet crystals with three different a-axis lattice spacing values ( $\sim 2.6$  Å,  $\sim 2.8$  Å, and  $\sim 3.0$  Å). These a-axis lattice spacing values are within  $\sim 10\%$  of the a-axis lattice spacing values previously reported for rhombohedral ( $x = 0.5$ ) and hexagonal ( $x = 0.61$  and  $0.67$ )  $K_x\text{CoO}_2$  phases.<sup>31,32</sup> The four distinct reciprocal lattice points in **Figure 3.1f** are located at three positions along the rotation axis around  $[001]$  indicating rotational misalignment of individual nanosheet crystals. The maximum angle of rotation between the reciprocal lattice points is  $\sim 5.7^\circ$  as visualized by the two yellow dashed lines in **Figure 3.1f**. Moiré fringes in the original crystal image (**Figure 3.1e**) also indicate a rotational misalignment ( $\alpha = 2 \cdot \arcsin(p/2D) = 5.5^\circ$ , where  $p$  is d-spacing and  $D$  is the distance between Moiré fringes<sup>39</sup>) close to the maximum rotational misalignment observed in the FFT image. Similar Moiré patterns were also observed for rotationally misaligned multilayer graphene superlattices by several other groups.<sup>39-41</sup> The rotational misalignment of the nanosheets combined with their perfect registry along the c-axis indicate that the nanosheets are stacked on top of each other with “turbostratic arrangement”. Such an arrangement was also previously observed for the stacking of  $\text{Na}_x\text{CoO}_2$  nanosheets.<sup>21</sup>

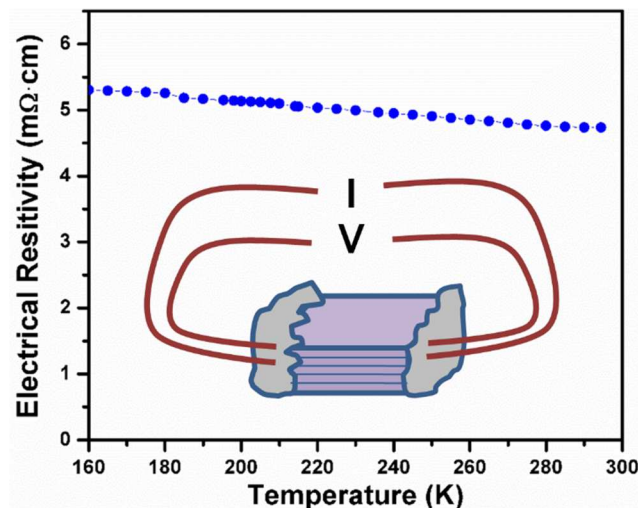


**Figure 3.1: TEM, HR-TEM and optical microscopy images of delaminated  $K_xCoO_{2.y}H_2O$  nanosheets.** (a) High (left) and low (right) magnification TEM images of delaminated nanosheets. (b) TEM images of thicker stacks of electron transparent delaminated nanosheets (c) TEM image (left) of a large delaminated nanosheet ( $\sim 45 \times 45 \mu m$ ) with optical microscopy image (right) of the same nanosheet under transmitted light. (d) TEM image (left) of a large, electron transparent

nanosheet ( $\sim 50 \times 20 \mu\text{m}$ ), ED pattern (middle) of the same nanosheet indicating that the nanosheet is crystalline and the orientation of cross-plane thickness is [001], dark-field image (right) confirming this ED pattern is representative of the entire nanosheet sheet and that the nanosheet is a single crystalline. (e) HR-TEM image of a delaminated nanosheet (red parallel lines indicate Moiré fringes forming due to the rotational misalignment of the nanosheets around [001] axis) (f) FFT of the image in (e) showing the reciprocal lattice of the delaminated nanosheet. The inset image in (f) shows four nearby reciprocal lattice points originating from nanosheets with different a-axis lattice spacings ( $\sim 2.6 \text{ \AA}$ ,  $\sim 2.8 \text{ \AA}$  and  $\sim 3.0 \text{ \AA}$ ) and rotational alignments. Two yellow dashed lines are drawn between the reciprocal lattice points of two rotationally misaligned nanosheets. The angle between the dashed lines indicates  $5.7^\circ$  of rotational misalignment which closely matches with  $5.5^\circ$  of rotational misalignment obtained from the Moiré fringes in (e). See **Figure B.3**, Appendix B, for more HR-TEM images of the delaminated nanosheets.

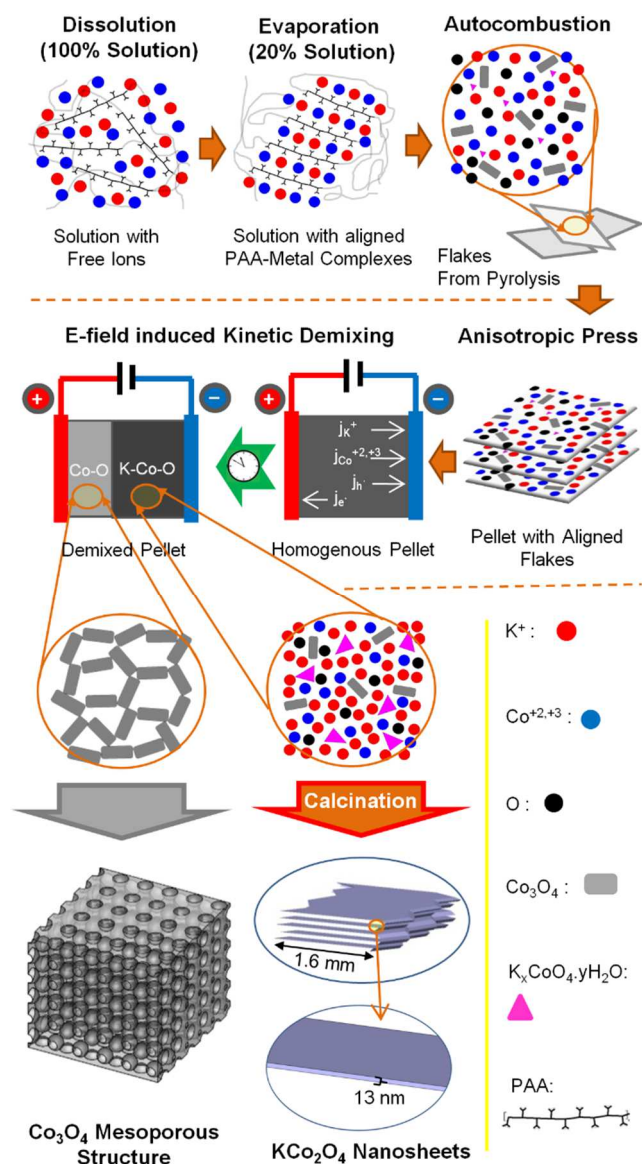
Electrical resistivity measurements were performed on the as-grown stacks of water stabilized  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets in a temperature range of 160 to 295 K. The two-point resistivity measurements were performed parallel to the nanosheets using two area contacts that are made on the sample with silver paint (See **Figure 3.2** inset drawing). The size of the nanosheet stack used in the measurement in **Figure 3.2** is  $730 \times 340 \times 6 \mu\text{m}$  ( $L \times W \times t$ ). The silver paint used for attaching the electrodes encroaches into the sample by up to  $\sim 290 \mu\text{m}$  in total from the two sides along the measurement axis resulting in an effective electrode separation of  $\sim 440 \mu\text{m}$ . In order

to obtain an upper limit for the in-plane resistivity value of the nanosheet stack, we assumed the electrode separation to be 440  $\mu\text{m}$ . This calculation should provide an overestimation for the in-plane resistivity ( $\rho_{ab}$ ) of the nanosheets for two more reasons: 1) The contact resistances at the interfaces between the silver paint and nanosheets are included in the two-point resistance measurement and 2) because the silver paint contacts are not perfectly affixed to only the surfaces of the  $\{100\}$  planes, the higher c-axis resistivity<sup>35</sup> will mix in with the lower in-plane values as has been seen previously in layered alkali cobaltates.<sup>26</sup> **Figure 3.2** shows the measured in-plane electrical resistivity values for a stack of water stabilized  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets. The single crystalline nature of the nanosheet stack along the measurement plane is critical for low electrical resistivity. As a comparison, polycrystalline samples measured with the same method result in resistivity values that are more than 2 orders of magnitude higher (See **Figure B.4**, Appendix B). At room temperature, the overestimated in-plane resistivity value of the  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets ( $\sim 4.7 \text{ m}\Omega \cdot \text{cm}$ ) is approximately two times higher compared to the bulk value previously reported by Tang *et al.* for  $\text{K}_{0.35}\text{CoO}_2 \cdot 0.34\text{H}_2\text{O}$  ( $\sim 2.3 \text{ m}\Omega \cdot \text{cm}$ ).<sup>35</sup> However, even with the overestimated in-plane resistivity values, the  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets have a remarkably high electrical conductivity for metal oxide nanosheets. To our knowledge, among the metal oxide nanosheets reported so far, these millimeter-length  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets have the highest electrical conductivity for metal oxide nanosheets longer than 10 microns. Overall, these nanosheets have the second highest electrical conductivity after  $\text{RuO}_2$  nanosheets.<sup>37,38</sup>



**Figure 3.2: Temperature dependent electrical resistivity values for  $K_xCoO_2 \cdot yH_2O$  nanosheets (blue dots).** Inset figure shows the configuration for the electrical resistivity measurements. The measurements were performed parallel to the nanosheets using two area contacts. See **Figure B.5**, Appendix B for optical microscopy and SEM images of the nanosheet stack used for the measurement.

The synthetic procedure for  $K_xCoO_2 \cdot yH_2O$  nanosheets is similar to the SGKD synthetic procedure for  $Na_xCoO_2$  nanosheets described in our previous work.<sup>21</sup> The nanosheet synthesis consists of a) the Pechini-method coordination of metal ions, b) pyrolysis into oxide flakes, c) pressurized pellet formation, d) electric-field (E-field) induced kinetic-demixing, and e) calcination (**Figure 3.3**).  $K_xCoO_2$  nanosheets are obtained after calcination. The samples are then soaked in water for controlled hydration of the  $K_xCoO_2$  nanosheets into the  $K_xCoO_2 \cdot yH_2O$  phase. Finally, free-standing nanosheets of  $K_xCoO_2 \cdot yH_2O$  are delaminated from the nanosheets stacks *via* ball-milling.

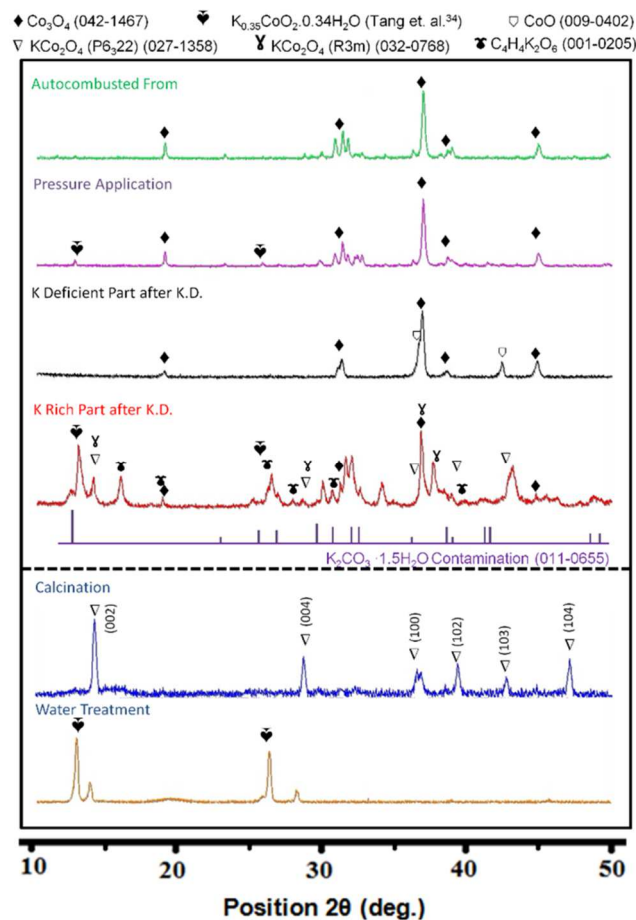


**Figure 3.3: Schematic diagram of the nano-manufacturing procedure for millimeter-length  $K_xCoO_2 \cdot yH_2O$  nanosheets.** Metal precursors and PAA are dissolved in water. The resulting solution is evaporated to  $\sim 80\%$  of its initial volume, leading to an increase in viscosity and formation of metal complexes. The solution is autocombusted into a pyrolyzed phase, which form as flakes. The pyrolyzed flakes are uniaxially pressed into a rectangular pellet. Electric current is applied to the pellet, and kinetic-demixing occurs due to the difference in the mobilities of K and Co atoms.



The K-deficient region is a porous network of  $\text{Co}_3\text{O}_4$  particles. The K-rich region is calcined at 1000 °C leading to the formation of the nano-layered structure due to anisotropic grain growth.

In the first step of the synthesis the Pechini method<sup>42</sup> is modified to entrap the metal ions in solutions by using PAA as the chelating agent. The metal cations are expected to be stabilized by the chelating groups on the ligand, through dipole forces between water molecules and metal ions, and by the crosslinking and physical tangling of polymer chains which can trap both the cations and solvents in a drying polymer-metal ion complex sheet.<sup>43-46</sup> The structure of these polymeric entrapped layers is evident after the solution is pyrolyzed. SEM images show <200 nm thick and <400  $\mu\text{m}$  long autocombusted flakes that include  $\text{Co}_3\text{O}_4$  crystals with a relatively minor amount of  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination according to the XRD pattern (**Figure 3.4** and **Figure B.6**, Appendix B). Due to the highly hygroscopic nature of K-(Co)-O compounds,  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination is observed in every step of the synthesis. To make measurements without the  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination, the K atoms in the samples were stabilized within the sample through controlled water treatment or controlled atmosphere XRD measurement (**Figure 3.4**).

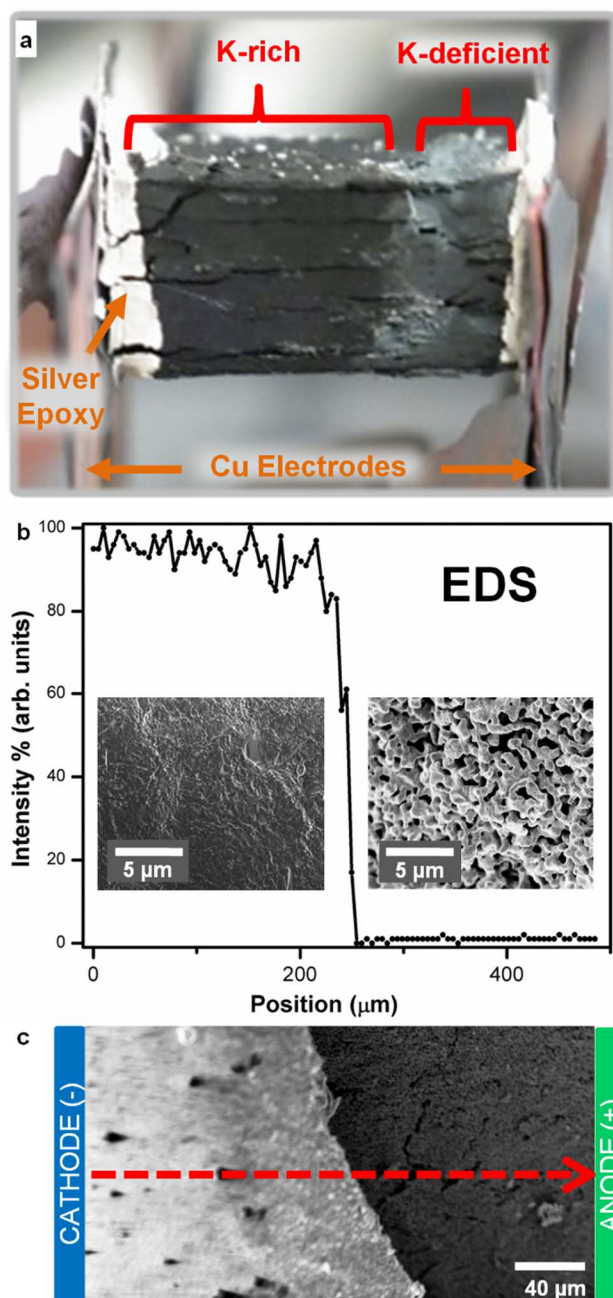


**Figure 3.4: X-ray powder diffraction for each step of the nanosheet synthesis procedure (ordered from top to bottom).** Pyrolyzed flakes after the autocombustion of the initial solution, anisotropically pressurized pellet of pyrolyzed flakes, K-deficient part of the pellet after E-field induced kinetic demixing, K-rich part of the pellet after E-field induced kinetic demixing, nanosheets after calcination of the K-rich part, water intercalated nanosheets after immersion of the calcined nanosheets in water. The samples in the form of pellets and nanosheet stacks were ground before the measurement. The peaks are identified for different phases with symbols and the available PDF numbers and/or references are listed for the phases in parentheses. Peaks without symbols are likely due to  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination forming on the

surface of K containing crystals. The reference peak positions for the  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination are shown separately with purple lines.

After the anisotropic pressure is applied to the autocombusted flakes by a rectangular die set, the XRD patterns indicate that the  $\text{Co}_3\text{O}_4$  phase formed during the autocombustion is still present (**Figure 3.4**). However, after the pressurization process a distinct and relatively low intensity peak appears that closely matches with the peak positions reported by Tang *et al.* for the  $\text{K}_{0.35}\text{CoO}_2 \cdot 0.34\text{H}_2\text{O}$  phase<sup>35</sup> (**Figure 3.4**). All of the peak intensities for both the autocombusted flakes and the pressurized pellet XRD are low, which can be due to existence of amorphous material(s) in the sample in addition to the crystalline material(s).

Performing E-field induced Kinetic-Demixing (**Figure 3.5a**) is the critical step to growing millimeter-length nanosheets. After application of a 500 mA current and a 1000 °C calcination, the nanosheet lateral lengths can reach ~2.1 millimeters. Samples with identical processing conditions but without kinetic-demixing do not contain nanosheets but arbitrarily shaped particles that reach only up to ~30  $\mu\text{m}$ , *i.e.*, 70-times shorter (See **Figure B.7**, Appendix B). E-field induced kinetic-demixing occurs 4-5 times faster in the direction perpendicular to the pressure axis compared to the direction parallel to the pressure axis, most likely because the pressure-aligned flakes provide a more contiguous pathway for the current than the cross-plane pathway provides.



**Figure 3.5: Characterization of the processing by electric-field induced kinetic-demixing.** (a) Optical image of the anisotropically pressurized pellet of pyrolyzed flakes after E-field induced kinetic-demixing. An electric current passes through Cu plates and contacts made of silver epoxy. The potassium-rich (K-rich) and K-deficient parts of the pellet can be visually distinguished: the K-deficient region is grey in color

and the K-rich region is black. (b) EDS line scan showing K concentration of the pellet after kinetic-demixing with respective SEM images of the K-deficient and K-rich regions. A stark contrast is seen between these regions: the K-deficient regions are more porous and contain nano-grains while the surface of the K-rich region is smooth. (c) SEM image showing cracking between the two regions. The separation line corresponds to the abrupt jump shown in the EDS of (b). Red dashed arrow shows direction of EDS scan.

The E-field application to the homogenous pellet results in an abrupt transition in K concentration along the E-field application axis and, according to XRD results, new phases form in the K-rich part of the pellet. Elemental analysis based on an SEM-EDS line scan shows the abrupt transition in K content and the morphology difference between the regions adjacent to the cathode and to the anode (**Figure 3.5b,c**). The region near the anode is K-depleted and extremely porous while the material near the cathode is K-rich and has a smooth unbroken surface (**Figure 3.5b** inset images, and **Figure B.8**, Appendix B). According to XRD, the K-deficient region is composed of only  $\text{Co}_3\text{O}_4$  and  $\text{CoO}$  crystals while the K-rich region includes K-(Co)-O based stoichiometric phases such as  $\text{KCo}_2\text{O}_4$  (both  $R3m$  and  $P6_322$ ),  $\text{K}_{0.35}\text{CoO}_2 \cdot 0.34\text{H}_2\text{O}$ , and  $\text{C}_4\text{H}_4\text{K}_2\text{O}_6$  in addition to  $\text{Co}_3\text{O}_4$  and  $\text{K}_2\text{CO}_3 \cdot 1.5\text{H}_2\text{O}$  contamination (**Figure 3.4**).

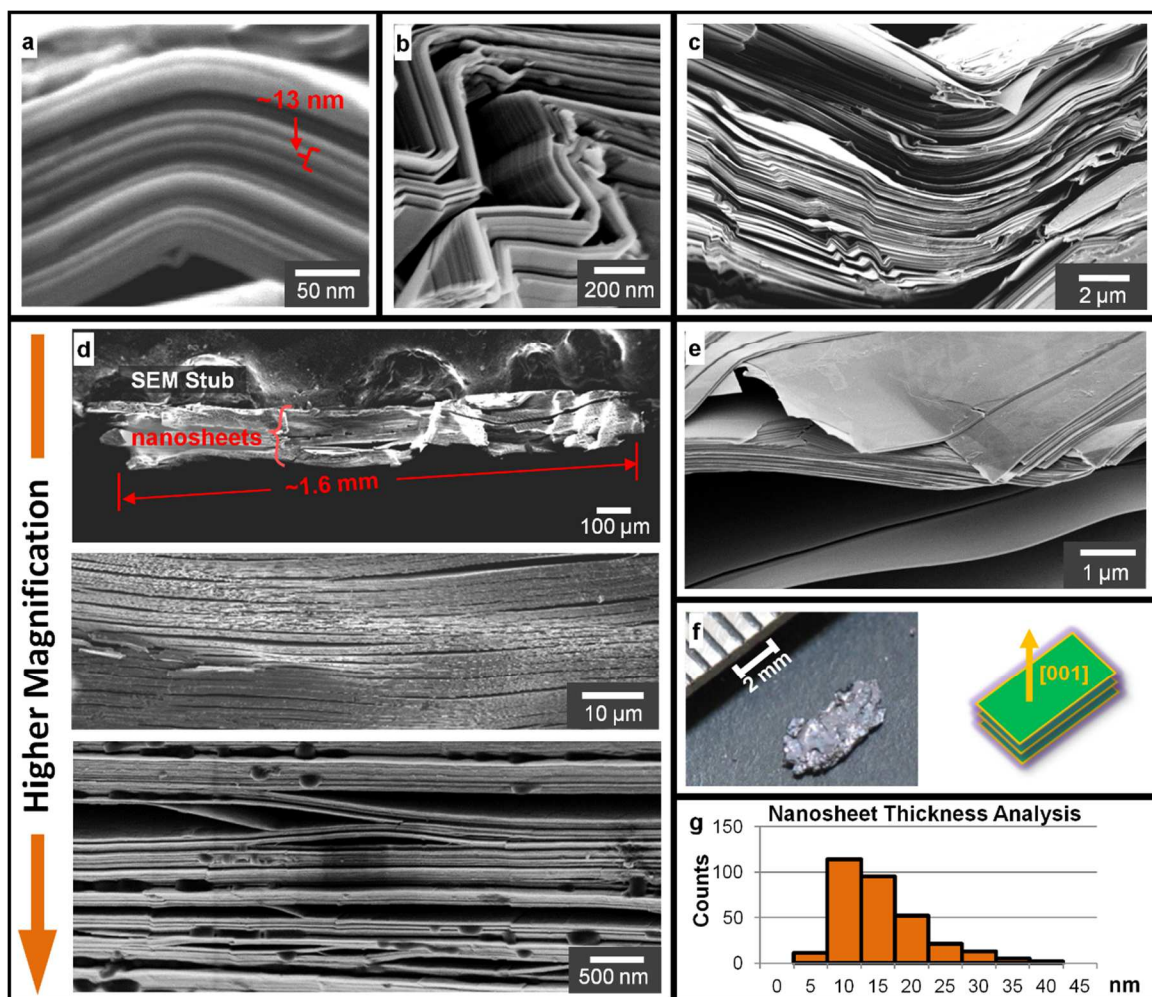
The kinetic-demixing process produces a K saturated compound, which increases the Co diffusion coefficient during the high temperature (1000 °C) calcination, leading to the millimeter-length  $\text{K}_x\text{CoO}_2$  nanosheets by anisotropic grain

growth. Correlating the EDS data with the sample volume suggests that the K:Co ratio can be as high as 1.8 in the K-rich side. At high temperatures ( $\sim 1000$  °C), this K-rich part of the pellet shows molten fluidity that is evident from the pellet shape deformation (*i.e.*, rectangular form into semi-spherical form) after the calcination (See **Figure B.9**, Appendix B). However, such shape deformation is not observed for the samples calcined directly after pressure application without the kinetic-demixing process (*i.e.*, when the K:Co ratio is maintained at 1). The fluidity in the K saturated compound at high temperatures is associated with a significant increase in the Co diffusion coefficient. This increase in the Co diffusion coefficient can be explained by the temperature dependence of the Co diffusion coefficient and also the “physical correlation effect”<sup>47</sup> between the Co and K. Due to the “physical correlation effect”, increasing the K content is expected to increase the diffusion coefficient of Co in the metal oxide mixture as previously proposed for a Na-(Co)-O mixture.<sup>21,48</sup> Our observation that high temperature molten fluidity exists only in kinetically-demixed samples (*i.e.*, if the samples were not kinetically-demixed then there was no molten fluidity) points to increased diffusion from the physical correlation of a larger K:Co ratio. The Co diffusion amplification significantly reduces kinetic limitations, and leads to long nanosheet formation through cooperative effects including: facile grain growth, fluidity in the pellet during calcination, and enhancement of the anisotropic growth along the (001) plane.

$\theta$ -2 $\theta$  XRD patterns of  $K_xCoO_2$  nanosheets obtained under inert atmosphere closely match with the  $KCo_2O_4$  (P6<sub>3</sub>22) phase, according to the XRD studies that were performed after decomposing the surface  $K_2CO_3 \cdot 1.5H_2O$  contamination. The XRD

pattern of the calcined samples closely matches with the hexagonal  $\text{KCo}_2\text{O}_4$  ( $\text{P6}_322$ ) phase (**Figure 3.4**). The two adjacent peaks observed around  $\sim 36.6^\circ$  indicate two different (100) lattice spacings (different by  $\sim 0.7\%$ ), while there is only one (002) lattice spacing according to the peak at  $14.2^\circ$ . Similar crystal structure was observed for  $\text{Na}_x\text{CoO}_2$  nanosheets with grazing incidence XRD measurements.<sup>21</sup>

SEM images of the metal oxide nanosheets are shown in **Figure 3.6**. A high magnification SEM image shows the typical  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheet thickness of  $\sim 13$  nm with a standard deviation of 6.4 nm (**Figure 3.6a,g**, for the histogram of over 300 nanosheets that are measured by SEM). The nanosheet lateral lengths are several millimeters (**Figure 3.6d** and **Figure B.10**, Appendix B, show two different stacks of nanosheets with lengths  $\sim 1.6$  mm and  $\sim 2.1$  mm, respectively). The nanosheets demonstrate extreme ductility: **Figure 3.6b** shows nano-scale bending of the nanosheets (bending radius  $\sim 5$  nm) and **Figure 3.6c** shows micro-scale bending of an 8-10  $\mu\text{m}$  thick stack of nanosheets (bending radius  $\sim 5$   $\mu\text{m}$ ). The nanosheets have a smooth surface (**Figure 3.6e**), implying a single crystalline nature. The nanosheet stacks (**Figure 3.6d**) consist of tens to hundreds of thousands of nanosheets, per stack. An optical image of a mechanically extracted nanosheet stack is shown in **Figure 3.6f**.



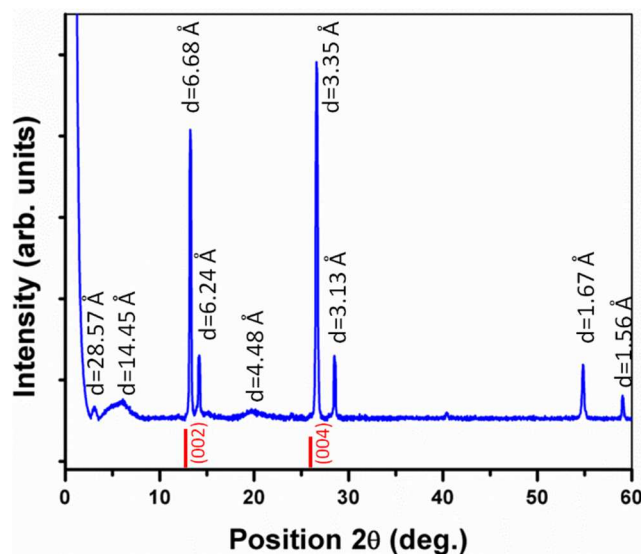
**Figure 3.6: SEM and optical images of stacked nanosheets.** (a) High magnification SEM image of the nanosheets showing the typical nanosheet thickness (b) SEM image of metal oxide nanosheets displaying extreme ductility after mechanical bending. (c) Bending of relatively thick stack of metal oxide nanosheets (8-10  $\mu\text{m}$ ) (d) SEM images of nanosheet stacks from low magnification (top) to high magnification (bottom). Total nanosheet length is 1.6 millimeters, and stack thickness is nominally 100 microns. (e) SEM image of slightly bent  $K_xCoO_2 \cdot yH_2O$  nanosheets showing smoothness of nanosheet surface. (f) (left) Optical image of a nanosheet stack obtained by mechanical extraction from the calcined bulk pellet. (right) The inset drawing



shows the alignment of the nanosheets in the nanosheet stack. (g) Histogram of nanosheet thickness obtained by SEM based thickness measurements (314 nanosheets were measured in total resulting in ~13 nm thickness in average)

After immersing the calcined samples in DI water for 5 days,  $\text{KCo}_2\text{O}_4$  nanosheets are converted to water stabilized  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets. Water treatment of the nanosheets resulted in peak shifts in the XRD pattern towards smaller angles, which corresponds to an increase in the c-axis lattice spacing (**Figure 3.4**). This increase in atomic layer separation with water interaction is similar to what has been observed for the  $\text{Na}_x\text{CoO}_2$  to  $\text{Na}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  transformation which occurs through the intercalation of water molecules between the two dimensional triangular  $\text{CoO}_2$  layers.<sup>27,49</sup> The two highest intensity peaks ( $2\theta = 13.1^\circ$  and  $2\theta = 26.4^\circ$ ) in the powder XRD pattern of the water stabilized nanosheets (**Figure 3.4**) closely match with the peak positions of (00L) planes reported by Tang *et al.* for the  $\text{K}_{0.35}\text{CoO}_2 \cdot 0.34\text{H}_2\text{O}$  phase.<sup>35</sup> The other two peaks positioned at slightly higher angles ( $2\theta = 14^\circ$  and  $2\theta = 28.3^\circ$ ), compared to the first pair, are most likely associated with another set of (00L) planes of a secondary  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  phase with lower water content. The formation of the secondary phase was repeatedly observed for many samples from different synthesis batches without significant alteration of the XRD peak positions of the (00L) planes belonging to the  $\text{K}_{0.35}\text{CoO}_2 \cdot 0.34\text{H}_2\text{O}$  phase and the secondary phase. Elemental analysis performed by WDS using reference samples (see Experimental Section) indicate that K:Co ratio for the water stabilized nanosheet stacks is ~ 0.44. This value is higher than the value found by Tang *et al.* through

SEM-EDS.<sup>34</sup> The discrepancy may arise from the type or lack of reference samples used for the Tang *et al.* EDS characterization. The absence of all XRD peaks other than the (00L) planes (**Figure 3.4**, bottom) is likely due to highly anisotropic nature of the  $K_xCoO_2 \cdot yH_2O$  crystals: with millimeter-length lateral sizes the nanosheet stacks show texturing through their alignment with the substrate and x-ray beam. Similar behavior was not observed in calcined samples because, unlike the water stabilized samples, the calcined samples were ground into arbitrary shaped submillimeter pieces without careful mechanical extraction of nanosheet stacks, resulting in a higher probability of random alignments for the crystals. **Figure 3.7** shows an XRD pattern of a single stack of water stabilized nanosheets when the x-ray scattering vector ( $Q$ ) is aligned perpendicular to the nanosheet stack. The peak positions in the XRD of the single stack of water stabilized nanosheets (**Figure 3.7**) match with the (00L) peak positions of the two different water intercalated phases observed in the powder XRD (**Figure 3.4**) of the same material. It is interesting that the two different phases are stable at the same time within a single stack of nanosheets. Considering the calculated d-spacing values, the peaks that appear at  $2\theta = 54.8^\circ$  and  $2\theta = 59^\circ$  in the XRD of the single stack of nanosheets are also associated with the same (00L) peaks observed in the powder XRD. On the other hand, the two wide, low intensity peaks at lower angles ( $2\theta = 3.1^\circ$  and  $2\theta = 6.1^\circ$ ) are likely associated with a separate set of (00L) peaks belonging to a third unknown phase. The peaks other than the (00L) peaks are absent because of texturing: the sample consists of a single nanosheet stack, and within the nanosheet stack the nanosheet crystals are well-aligned on top of each other.

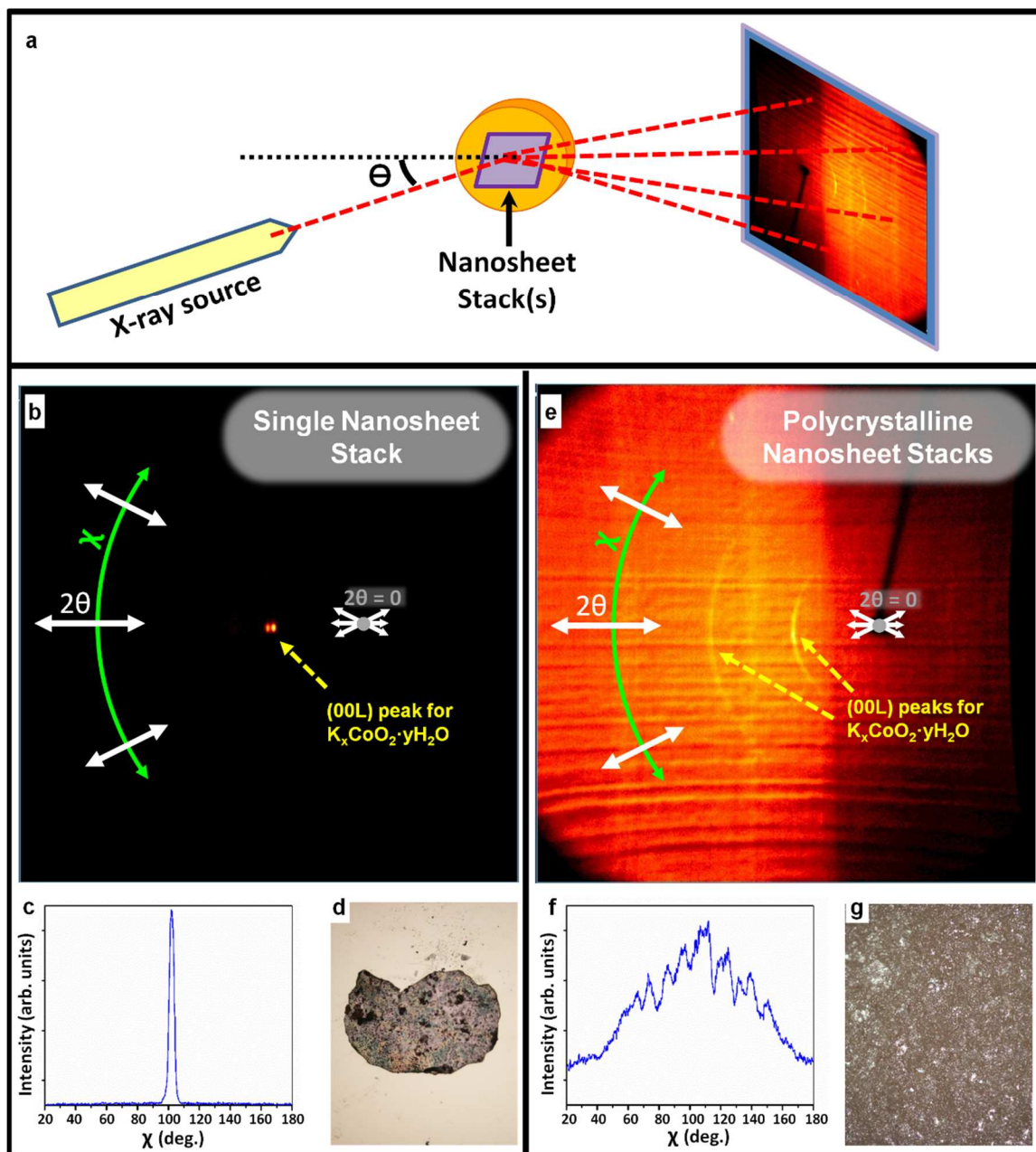


**Figure 3.7: XRD of a single stack of water intercalated nanosheets.** The stacking axis of the nanosheets were aligned parallel to X-ray scattering vector ( $Q \parallel c$ -axis). D-spacing values are indicated for all peaks. The reference peak positions for the  $K_{0.35}CoO_2 \cdot 0.34H_2O$  phase are shown with red stick lines.<sup>34</sup> Appearance of only (00L) peaks in the XRD pattern indicates that the nanosheets are large crystals and that they are strongly oriented (textured in alignment with respect to the incoming beam).

XRD studies performed using an area detector also indicate that nanosheet crystals are well-aligned along their c-axes within the nanosheet stacks (**Figure 3.8**). In the area detector XRD studies the angle between the sample stage and the incoming linear x-ray beam is tuned such that x-rays incline to the sample surface at  $\theta = 6.55^\circ$  (**Figure 3.8a**). The measurements are performed on a single nanosheet stack sample and, for comparison, on a polycrystalline sample. The single nanosheet stack sample is oriented on the sample stage such that the nanosheet stack top surface is parallel to the sample stage (i.e., the nanosheet stacking axis is perpendicular to the sample

stage). The polycrystalline sample is prepared by uni-axially pressing many randomly oriented nanosheet stacks into a rectangular pellet (see **Figure B.4**, Appendix B caption). Area detector XRD plots for the single nanosheet stack sample and the polycrystalline nanosheet sample are shown in **Figure 3.8b** and **3.8e**, respectively, with the  $2\theta$  and  $\chi$  axes marked. Distinct peaks along  $2\theta$  indicate diffraction from different lattice spacings and different peaks along  $\chi$  indicate different crystal orientations. The spread of (00L) peaks along the  $\chi$  axis in **Figure 3.8b** and **3.8e** is an indication of different crystal orientations within the sample. The (00L) peak for the single nanosheet stack shows negligible spread along the  $\chi$  axis (FWHM =  $\sim 3.6^\circ$ ) indicating that the nanosheet crystals are well-aligned on top of each other within the nanosheet stack and the nanosheet stack does not consist of randomly oriented nanosheets (**Figure 3.8c**). However, the large  $\chi$  axis width (FWHM =  $\sim 78.4^\circ$ ) of the (00L) peak for the polycrystalline nanosheet sample indicates that that the polycrystalline nanosheet sample consists numerous nanosheets that are oriented along different directions (**Figure 3.8f**). The single nanosheet stack sample produces (00L) peaks only within close proximity of  $2\theta = 13.1^\circ$  because the incident x-ray beam inclines to nanosheet crystals at  $\theta = 6.55^\circ$  and the c-axes of all of the nanosheet crystals are well-aligned in the sample with negligible deviation. However, in the polycrystalline nanosheet sample (00L) peaks are observed at both  $2\theta = 13.1^\circ$  and  $2\theta = 26.4^\circ$  because the x-ray beam inclines to some of the randomly oriented crystals within the polycrystalline sample at  $6.55^\circ$  and some others at  $13.2^\circ$  satisfying the diffraction condition for both two  $2\theta$  peaks at  $13.1^\circ$  and  $26.4^\circ$ . Background noise is more apparent in the area detector XRD plot of the polycrystalline nanosheet sample

because the signal to noise ratio is significantly lower for the polycrystalline nanosheet sample compared to the single nanosheet stack sample, as expected for XRD signals of polycrystalline material compared to single crystal. XRD peaks from other crystal planes are not seen in the polycrystalline sample likely because of this poor signal to noise ratio. Optical images for both the single nanosheet stack sample and the polycrystalline nanosheet sample are shown in **Figure 3.8d** and **3.8g**, respectively (only top surface can be seen in the image). The top surface of the single nanosheet stack has excessive dents and scratches mostly due to mechanical handling during the extraction of the nanosheet stack from the calcined pellet. The top surface of the polycrystalline sample shows multiple grains.



**Figure 3.8: XRD analysis of  $K_xCoO_2 \cdot yH_2O$  nanosheet stack(s) using area detector:** (a) Simplified schematics of the XRD setup (See Supporting Information Methods 1). X-rays incline to the sample surface at  $\theta = 6.55^\circ$  and area detector is oriented to detect diffraction ( $2\theta$ ). The spot size of the x-ray beam is large enough to cover entire surface of the single nanosheet stack sample ( $L \times W \approx 0.8 \times 0.5$  mm). The

polycrystalline nanosheet sample is significantly larger than the post size of the x-ray beam. (b) Area detector XRD plot from the single nanosheet stack mounted on Si wafer. (c) The intensity of the  $2\theta = 13.1^\circ$  (00L) peak from the  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheet stack is plotted against  $\chi$ . The (00L) peak from the single nanosheet stack sample has negligible width (FWHM =  $\sim 3.6^\circ$ ) along the  $\chi$  axis (instrumental broadening along  $\chi$  is  $\sim 1.55^\circ$ ). (e) Area detector XRD plot from the polycrystalline nanosheet sample. (f) The intensity for the  $2\theta = 13.1^\circ$  (00L) peak from the polycrystalline  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheet sample is plotted against  $\chi$ . The FWHM of the (00L)  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  peak from the polycrystalline nanosheet sample is  $\sim 78.4^\circ$ . (d) and (e) Optical images of the single nanosheet stack and the polycrystalline nanosheet samples used in the area detector XRD experiment.

### 3.4. Conclusion

Electrically conductive, millimeter-length nanosheets of  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  were produced for the first time using the SGKD method. In-plane electrical resistivity of the  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets is less than  $\sim 4.7 \text{ m}\Omega \cdot \text{cm}$  at room temperature, which corresponds to the highest electrical conductivity in the literature for metal oxide nanosheets longer than 10 microns. The final product consists of tens of thousands of stacked nanosheets with extreme ductility, which makes a convenient form for device integration. The average thicknesses of the millimeter size nanosheets is  $\sim 13 \text{ nm}$  resulting in a very high anisotropic aspect ratio (50,000: 50,000:1). The nanosheets can be easily mechanically bent, displaying their extreme ductility. The nanosheets are readily delaminated into free-standing sheets that are tens of microns long. SAED

studies indicated that the crystal properties are not altered by the delamination process. According to the HR-TEM characterization of delaminated nanosheets, the nanosheets are turbostratically stacked on top of each other. It should be able to synthesize nanosheets of other atomically layered oxides with this method.

## REFERENCES

- <sup>1</sup> M. Aksit, B. C. Hoselton, H. J. Kim, D. H. Ha, and R. D. Robinson, *ACS Appl. Mater. Interfaces* **5** (18), 8998 (2013).
- <sup>2</sup> X. Chen, N. Q. Zhang, and K. N. Sun, *J. Phys. Chem. C* **116** (40), 21224 (2012).
- <sup>3</sup> T. Nakato, K. Nakamura, Y. Shimada, Y. Shido, T. Houryu, Y. Iimura, and H. Miyata, *J. Phys. Chem. C* **115** (18), 8934 (2011).
- <sup>4</sup> M. Osada and T. Sasaki, *J. Mater. Chem.* **19** (17), 2503 (2009).
- <sup>5</sup> J. N. Coleman, M. Lotya, A. O'Neill, S. D. Bergin, P. J. King, U. Khan, K. Young, A. Gaucher, S. De, R. J. Smith, I. V. Shvets, S. K. Arora, G. Stanton, H. Y. Kim, K. Lee, G. T. Kim, G. S. Duesberg, T. Hallam, J. J. Boland, J. J. Wang, J. F. Donegan, J. C. Grunlan, G. Moriarty, A. Shmeliov, R. J. Nicholls, J. M. Perkins, E. M. Grieveson, K. Theuwissen, D. W. McComb, P. D. Nellist, and V. Nicolosi, *Science* **331** (6017), 568 (2011).
- <sup>6</sup> R. Ma and T. Sasaki, *Adv. Mater.* **22** (45), 5082 (2010).
- <sup>7</sup> A. Takagaki, M. Sugisawa, D. Lu, J. N. Kondo, M. Hara, K. Domen, and S. Hayashi, *J Am Chem Soc* **125** (18), 5479 (2003).
- <sup>8</sup> L. Z. Wang, K. Takada, A. Kajiyama, M. Onoda, Y. Michiue, L. Q. Zhang, M. Watanabe, and T. Sasaki, *Chem Mater* **15** (23), 4508 (2003).



- <sup>9</sup> R. Mas-Balleste, C. Gomez-Navarro, J. Gomez-Herrero, and F. Zamora, *Nanoscale* **3** (1), 20 (2011).
- <sup>10</sup> S. Z. Butler, S. M. Hollen, L. Cao, Y. Cui, J. A. Gupta, H. R. Gutierrez, T. F. Heinz, S. S. Hong, J. Huang, A. F. Ismach, E. Johnston-Halperin, M. Kuno, V. V. Plashnitsa, R. D. Robinson, R. S. Ruoff, S. Salahuddin, J. Shan, L. Shi, M. G. Spencer, M. Terrones, W. Windl, and J. E. Goldberger, *Acs Nano* **7** (4), 2898 (2013).
- <sup>11</sup> C. Tagusagawa, A. Takagaki, S. Hayashi, and K. Domen, *J. Phys. Chem. C* **113** (18), 7831 (2009).
- <sup>12</sup> L. Zhang, X. Tang, and W. Gao, *Cryst. Growth Des.* **8** (7), 2489 (2008).
- <sup>13</sup> W. W. Sun, F. Cao, Y. M. Liu, X. Z. Zhao, X. G. Liu, and J. K. Yuan, *J. Mater. Chem.* **22** (39), 20952 (2012).
- <sup>14</sup> S. Suzuki and M. Miyayama, *J Electrochem Soc* **154** (5), A438 (2007).
- <sup>15</sup> S. Suzuki and M. Miyayama, *J. Phys. Chem. B* **110** (10), 4731 (2006).
- <sup>16</sup> M. Yeager, W. X. Du, R. Si, D. Su, N. Marinkovic, and X. W. Teng, *J. Phys. Chem. C* **116** (38), 20173 (2012).
- <sup>17</sup> M. Osada, Y. Ebina, H. Funakubo, S. Yokoyama, T. Kiguchi, K. Takada, and T. Sasaki, *Adv. Mater.* **18** (8), 1023 (2006).
- <sup>18</sup> M. Osada, G. Takanashi, B. W. Li, K. Akatsuka, Y. Ebina, K. Ono, H. Funakubo, K. Takada, and T. Sasaki, *Adv. Funct. Mater.* **21** (18), 3482 (2011).
- <sup>19</sup> M. Osada, K. Akatsuka, Y. Ebina, H. Funakubo, K. Ono, K. Takada, and T. Sasaki, *Acs Nano* **4** (9), 5225 (2010).
- <sup>20</sup> Y. Masuda, Y. Hamada, W. S. Seo, and K. Koumoto, *J. Nanosci.*

- Nanotechnol. **6** (6), 1632 (2006).
- <sup>21</sup> M. Aksit, D. P. Toledo, and R. D. Robinson, *J. Mater. Chem.* **22** (13), 5936 (2012).
- <sup>22</sup> I. Terasaki, Y. Sasago, and K. Uchinokura, *Phys. Rev. B* **56** (20), 12685 (1997).
- <sup>23</sup> Y. Y. Wang, N. S. Rogado, R. J. Cava, and N. P. Ong, *Nature* **423** (6938), 425 (2003).
- <sup>24</sup> L. W. Shacklette, T. R. Jow, and L. Townsend, *J Electrochem Soc* **135** (11), 2669 (1988).
- <sup>25</sup> M. L. Foo, Y. Y. Wang, S. Watauchi, H. W. Zandbergen, T. He, R. J. Cava, and N. P. Ong, *Phys. Rev. Lett.* **92** (24), 247001 (2004).
- <sup>26</sup> M. Lee, L. Viciu, L. Li, Y. Y. Wang, M. L. Foo, S. Watauchi, R. A. Pascal, R. J. Cava, and N. P. Ong, *Nat. Mater.* **5** (7), 537 (2006).
- <sup>27</sup> Kazunori Takada, Hiroya Sakurai, Eiji Takayama-Muromachi, Fujio Izumi, Ruben A. Dilanian, and Takayoshi Sasaki, *Nature* **422** (6927), 53 (2003).
- <sup>28</sup> W. Liu and M. Asheghi, *J. Appl. Phys.* **98** (12), 123523 (2005).
- <sup>29</sup> Y. J. Wang, D. P. Wilkinson, and J. J. Zhang, *Chem. Rev.* **111** (12), 7625 (2011).
- <sup>30</sup> M. Pollet, M. Blangero, J. P. Doumerc, R. Decourt, D. Carlier, C. Denage, and C. Delmas, *Inorg. Chem.* **48** (20), 9671 (2009).
- <sup>31</sup> C. Delmas, C. Fouassier, and P. Hagenmuller, *J. Solid State Chem.* **13** (3), 165 (1975).
- <sup>32</sup> M. Pollet, M. Blangero, J. P. Doumerc, R. Decourt, D. Carlier, C. Denage, and

- C. Delmas, *Inorg. Chem.* **48** (20), 9671 (2009).
- <sup>33</sup> S. Nakamura, J. Ohtake, N. Yonezawa, and S. Iida, *J. Phys. Soc. Jpn.* **65** (2), 358 (1996).
- <sup>34</sup> J. Sugiyama, Y. Ikedo, P. L. Russo, H. Nozaki, K. Mukai, D. Andreica, A. Amato, M. Blangero, and C. Delmas, *Phys. Rev. B* **76** (10), 104412 (2007).
- <sup>35</sup> H. Y. Tang, H. Y. Lin, M. J. Wang, M. Y. Liao, J. L. Liu, F. C. Hsu, and M. K. Wu, *Chem Mater* **17** (8), 2162 (2005).
- <sup>36</sup> M. Osada and T. Sasaki, *Adv. Mater.* **24** (2), 210 (2012).
- <sup>37</sup> J. Sato, H. Kato, M. Kimura, K. Fukuda, and W. Sugimoto, *Langmuir* **26** (23), 18049 (2010).
- <sup>38</sup> D. S. Kim, T. C. Ozawa, K. Fukuda, S. Ohshima, I. Nakai, and T. Sasaki, *Chem Mater* **23** (11), 2700 (2011).
- <sup>39</sup> D. L. Miller, K. D. Kubista, G. M. Rutter, M. Ruan, W. A. de Heer, P. N. First, and J. A. Stroscio, *Phys. Rev. B* **81** (12), 125427 (2010).
- <sup>40</sup> L. Brown, R. Hovden, P. Huang, M. Wojcik, D. A. Muller, and J. Park, *Nano Lett.* **12** (3), 1609 (2012).
- <sup>41</sup> C. C. Lu, Y. C. Lin, Z. Liu, C. H. Yeh, K. Suenaga, and P. W. Chiu, *Acs Nano* **7** (3), 2587 (2013).
- <sup>42</sup> M. P. Pechini, U.S. Patent No. 3,330,697 (1967).
- <sup>43</sup> M. Kakihana, *J. Sol-Gel Sci. Technol.* **6** (1), 7 (1996).
- <sup>44</sup> M. A. Gulgun, M. H. Nguyen, and W. M. Kriven, *J Am Ceram Soc* **82** (3), 556 (1999).
- <sup>45</sup> M. A. Gulgun, W. M. Kriven, and M. H. Nguyen, US Patent No. 6,482,387

(2002).

- <sup>46</sup> M. H. Nguyen, S. J. Lee, and W. M. Kriven, *Journal of Materials Research* **14** (8), 3417 (1999).
- <sup>47</sup> J. Philibert, *Atom movements: diffusion and mass transport in solids*. (Éditions de Physique, 1991).
- <sup>48</sup> M. Schnehage, R. Dieckmann, and H. Schmalzried, *Berichte Der Bunsen-Gesellschaft-Physical Chemistry Chemical Physics* **86** (11), 1061 (1982).
- <sup>49</sup> R. E. Schaak, T. Klimczuk, M. L. Foo, and R. J. Cava, *Nature* **424** (6948), 527 (2003).

## CHAPTER 4

### 4. MISFIT LAYERED $\text{Ca}_3\text{Co}_4\text{O}_9$ AS A HIGH FIGURE OF MERIT p-TYPE TRANSPARENT CONDUCTING OXIDE FILM THROUGH SOLUTION PROCESSING \*

#### **4.0. Abstract**

$\text{Ca}_3\text{Co}_4\text{O}_9$  thin films synthesized through solution processing are shown to be high-performing, p-type TCOs. The synthesis method is a cost-effective and scalable process that consists of sol-gel chemistry, spin coating, and heat treatments. The process parameters can be varied to produce TCO thin films with sheet resistance as low as  $5.7 \text{ k}\Omega/\text{sq}$  ( $\rho \approx 57 \text{ m}\Omega\cdot\text{cm}$ ) or with average visible range transparency as high as 67%. The most conductive  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin film has near infrared region optical transmission as high as 85%. The FOM for the top-performing  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin film ( $151 \text{ M}\Omega^{-1}$ ) is higher than FOM values reported in the literature for all other solution processed, p-type TCO thin films and higher than most others prepared by PVD and CVD. Transparent conductivity in misfit layered oxides presents new opportunities for TCO compositions.<sup>1</sup>

#### **4.1. Introduction**

One of the most essential components of devices such as solar cells, LCDs, OLEDs, and touch screens are electrical contacts having transparency to visible light. The most commonly used materials for such electrical contacts are heavily doped,

---

\* Originally Published as: Mahmut Aksit, Sanjeev K. Kolli, Ian M. Sluach and Richard D. Robinson, "Misfit layered  $\text{Ca}_3\text{Co}_4\text{O}_9$  as a high figure of merit p-type transparent conducting oxide film through solution processing", *Applied Physics Letters* 104, 161901 (2014). Reprinted with Permission from American Institute of Physics.

large band-gap metal oxide semiconductors with n-type charge carriers (e.g., ITO).<sup>2,3</sup> In contrast to the widespread use of n-type TCOs, p-type TCOs have not been commercialized yet due to their significantly lower carrier mobility and electrical conductivity compared to n-type TCOs.<sup>4-6</sup> High conductivity p-type TCOs could serve as critical components for various technological developments such as efficient charge injection layers for organic light emitting devices,<sup>6</sup> solar cells with better band matching current collectors,<sup>7,8</sup> invisible circuits,<sup>6,9</sup> and applications in near infrared optoelectronics where n-type TCOs provide poor optical transmission.<sup>10</sup> Since the pioneering work on thin films of delafossite  $\text{CuAlO}_2$ ,<sup>11</sup> p-type TCOs have been vigorously researched with the goal of creating invisible circuits.<sup>4,9,12-15</sup> To date, the materials with the highest conductivity among p-type TCOs is  $\text{CuCr}_{1-x}\text{Mg}_x\text{O}_2$ , with conductivity of 220 S/cm and visible range transparency of 30-40%.<sup>16</sup> However, thin film manufacturing routes in the majority of these studies have been restricted to CVD and PVD techniques. These methods are more expensive and less scalable than solution-based techniques, which are simpler and faster.<sup>6</sup> Solution-based synthesis techniques, so far, have provided only limited success for p-type TCO thin films, with performance hampered by low conductivity, induced from non-uniformity and excessive porosity.<sup>9,17-19</sup> Until now, the highest p-type conductivity from solution-based techniques is only  $\sim 1$  S/cm.<sup>19</sup>

In this work, a scalable and cost-effective manufacturing technique is reported for nanostructured, p-type TCO thin films of  $\text{Ca}_3\text{Co}_4\text{O}_9$ . Homogenous, p-type TCO thin films with 5.7 k $\Omega$ /sq sheet resistance ( $R_s$ ) and  $\sim 100$  nm thickness ( $t$ ) (corresponding to a conductivity of  $\sigma = 1/(R_s t) \approx 18$  S/cm) can be obtained by our

simple method that is based on sol-gel chemistry and spin coating. The average visible range optical transparency for the films can be varied from 31% to 67%, with a concomitant change in conductivity. We observed very high optical transmission in the near infrared region, reaching up to 85% for our most conductive TCO film.

$\text{Ca}_3\text{Co}_4\text{O}_9$  is a misfit-layered oxide with two alternating monoclinic subsystems (rock salt-type  $\text{Ca}_2\text{CoO}_3$  and  $\text{CdI}_2$ -type  $\text{CoO}_2$ ) that have identical lattice parameters for the  $a$  and  $c$  axes but different lattice parameters for the  $b$  axis.<sup>20</sup> Although  $\text{Ca}_3\text{Co}_4\text{O}_9$  has been very well known as a remarkable p-type thermoelectric material, with properties such as high in-plane conductivity ( $\sigma_{ab} > 500 \text{ S/cm}$ ) and Seebeck coefficient ( $S_{ab} > 120 \mu\text{V K}^{-1}$ ),<sup>20,21</sup> superior optoelectronic properties of  $\text{Ca}_3\text{Co}_4\text{O}_9$  have not been reported previously, and transparent conductivity has not been observed in misfit layered oxides.

## **4.2. Methods**

### **4.2.0. Synthesis of p-type TCO Thin Films**

The synthetic method for the transparent conducting thin films of calcium cobalt oxide employs the Pechini method, in which an organic chelating agent dissolves metal precursors in an appropriate solvent. The resulting homogenous liquid solution is then evaporated into a viscous resin that is spun coated on quartz substrates prior to in-furnace calcination. We modified the Pechini method to polymerically entrap the metal ions in aqueous solutions by using PAA (average molecular weight =  $M_w \sim 1800$ ) as the chelating agent.<sup>22</sup> Appropriate quantities of PAA, cobalt(II) nitrate hexahydrate and calcium nitrate tetrahydrate are dissolved in deionized water at room temperature. Concentrations of each of the metal salts in the solution are 0.205 M,

providing a Ca to Co ratio of 1:1. The ratio of PAA carboxylate groups to total metal ions is 2:1, resulting in a total solute concentration of 1.23 M, considering the number of PAA monomers and metals salts in the solution. The solution is evaporated at 150°C with continuous stirring until it reaches the desired solute concentration, forming a viscous resin. As the solution is evaporated, chelating groups on the ligand stabilize the metal cations in the solution through dipole forces between water molecules and metal ions, and by the physical tangling and crosslinking of polymer chains which can trap both the solvent and cations in a drying sheet of polymer-metal ion complex.<sup>22,23</sup> The solute concentration in the resin is adjusted by controlling the evaporated solvent volume. Evaporating the solution to 50%, 40%, 35%, and 30% of the initial solution volume results in total solute concentrations of 2.5, 3.1, 3.5 and 4.1 M ( $\pm$  3% error), respectively. The evaporated resin is spin coated on 1 inch diameter polished quartz substrates at 6000 rpm. The solution is injected from a syringe in a continuous stream for <2 seconds onto the quartz substrate. Spin rates lower than 6000 rpm result in notably more inhomogeneous resin coatings in terms of transparency and color. In order to eliminate the water content and organic species in the coated films, the samples are baked under vacuum at 80 and 150 °C consecutively for 2 hours each and then in air at 150 °C for another 2 hours. Setting the initial baking temperature to 80 °C prevents rapid outgassing of the water content preventing bubble and crack formation on the coated films. Similarly, the vacuum environment reduces the burning rate of the organic content resulting in slower outgassing of combustion products. The vacuum environment also speeds the time required for the low temperature outgassing of water. In the second step, the temperature is raised to



150 °C because this temperature is typically needed for outgassing of C content from PAA and N content from metal salts.<sup>24</sup> Introducing air in the last baking step ensures complete burning and outgassing of the remaining N and C content. The resin films are then calcined at 650 °C in a preheated box furnace for 5, 15, 25, or 35 minutes. The samples are covered by crucibles during calcination in order to avoid undesired radiative heating of the sample surface.

#### ***4.2.1. Measurement and Evaluation of Transparent Conductivity***

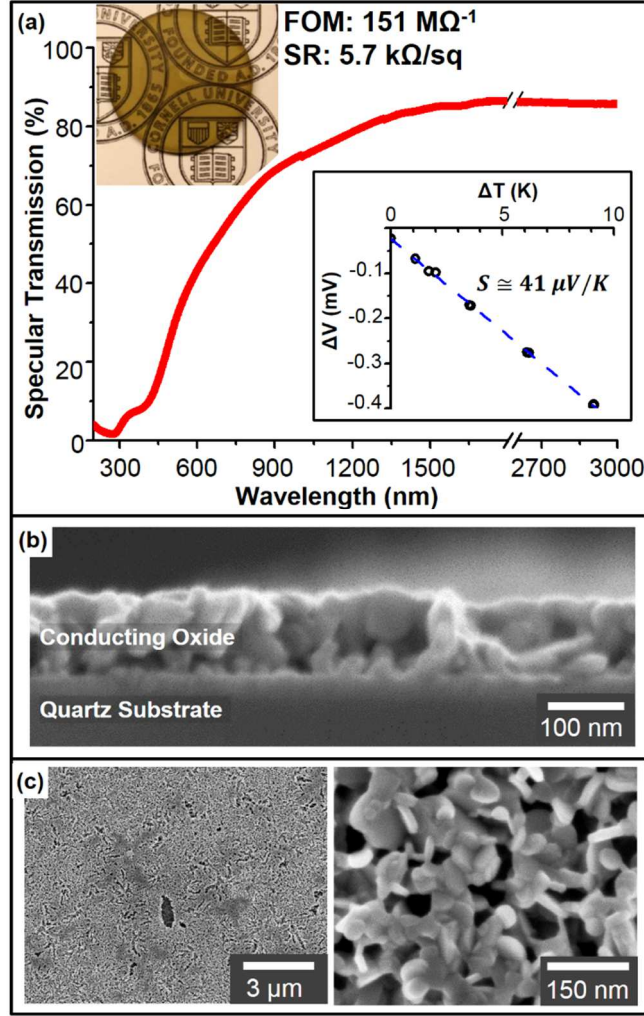
Specular optical transmissions and sheet resistances of calcium cobalt oxide thin films are measured to evaluate the TCO properties. A linear array, four-point probe is used for sheet resistance measurements. Sheet resistances of TCO films are determined by averaging 12 equally spaced measurements along two perpendicular lines passing through the center of the circular sample. The measurements are limited to locations that are within 8 mm from the center of the sample to avoid edge effects. Optical transmission measurements are performed using a Shimadzu UV3600 UV-Vis spectrometer and an Ocean Optics USB2000+VIS-NIR spectrometer, with an uncoated quartz disc used as a reference for these measurements. The optical transmission is measured through a ~5 x 15 mm rectangular area close to the center of the sample. Optical transmission though two different areas on the same sample are typically within 1% of each other. Because the high optical transmission and electrical conductivity are conflicting properties, the transparent conducting films are evaluated based on their FOM  $F = -1/(R_s \times \ln T)$  where  $R_s$  is the sheet resistance and  $T$  is the optical transmission.<sup>17</sup> We calculated FOM based on optical transmission in the visible range by averaging transmission values at photon energies of 1.77, 2,

2.25, 2.5, 2.75, and 3 eV.<sup>17</sup>

### 4.3. Results and Discussion

The highest FOM is found for the TCO thin films of calcium cobalt oxide when a 3.5 M total solute concentration resin is used and the film is calcined for 25 minutes at 650 °C. **Figure 4.1a** shows the specular optical transmission of the high FOM thin film between photon wavelengths of 200 and 3000 nm. The optical transmission increases with increasing photon wavelength ranging from 8.9% to 53.7% at the lower and higher edges of the visible range (400 and 700 nm, respectively). The optical transmission is significantly enhanced in the near infrared region, reaching up to 85% for photon wavelengths longer than ~1.5 μm. The film has visibly homogenous transparency with translucent brown color (**Figure 4.1a**, top left inset). The average photon transmission in the visible range and the sheet resistance ( $R_s$ ) for the film are 31.3% and 5.7 kΩ/sq, respectively, resulting in a FOM of 151 MΩ<sup>-1</sup>. Voltage and temperature differences ( $\Delta V$  and  $\Delta T$ ) are measured between two points on the thin film at room temperature in order to find the Seebeck coefficient (**Figure 4.1a**, bottom right inset). The temperature difference between the two points is induced by a resistive heater at the hot point while the cold point remains at room temperature. The data points in the  $\Delta V$  vs.  $\Delta T$  plot fit to a line (**Figure 4.1a**, bottom right inset, open black circles and dashed blue line, respectively), with a negative slope indicating a positive Seebeck coefficient of  $S = -\frac{\Delta V}{\Delta T} \cong 41 \mu V/K$  and p-type conductivity. The FOM value for the p-type TCO thin film in **Figure 4.1** is significantly higher than the FOM values of all other solution processed p-type TCO thin films previously reported and even higher than FOM values of most PVD and

CVD prepared films.<sup>17</sup> The thickness of the TCO thin film is ~100 nm according to a cross-sectional SEM (**Figure 4.1b** and **Figure C.1a**, Appendix C) and 106 nm as measured by contact profilometry.<sup>25</sup> The contact profilometry measurements performed on the film surface over 1 mm length with 1  $\mu\text{m}$  lateral resolution indicate that the standard deviation in the film height is 16 nm. Surface SEM images show that the film consists of nano-porous network of ~50–100 nm long nano-plates (**Figure 4.1c**, right) and the film is mostly smooth and homogenous with occasional micron-scale cracks (**Figure 4.1c**, left and **Figure C.1b**, Appendix C).<sup>25</sup> These micron-scale cracks form during calcination, mostly likely due to grain growth by coalescence of crystals. Assuming ~100 nm thickness, the electrical conductivity of the TCO thin film can be calculated as  $\sigma = 1/(R_{st}) \approx 18 \text{ S/cm}$  (resistivity,  $\rho \approx 57 \text{ m}\Omega \text{ cm}$ ). This conductivity is more than 25 times lower compared to in-plane conductivities previously reported for single crystalline  $\text{Ca}_3\text{Co}_4\text{O}_9$  ( $>500 \text{ S/cm}$ ).<sup>20,21</sup> The likely reason for the reduction in conductivity is the nanoporous and nanocrystalline microstructure of the film that reduces the effective cross-sectional area for electrical conduction and introduces contact resistance between the  $\text{Ca}_3\text{Co}_4\text{O}_9$  crystals. The micron scale cracks in the film can also reduce the specular optical transmission due to scattering of light, therefore, elimination of such defects should improve the optical transmission and FOM of the film. The nanoporous and nanocrystalline microstructure of the film is unlikely to cause significant optical scattering because the pore and crystal sizes are significantly shorter than the visible light wavelengths (**Figure 4.1c**).<sup>26</sup>

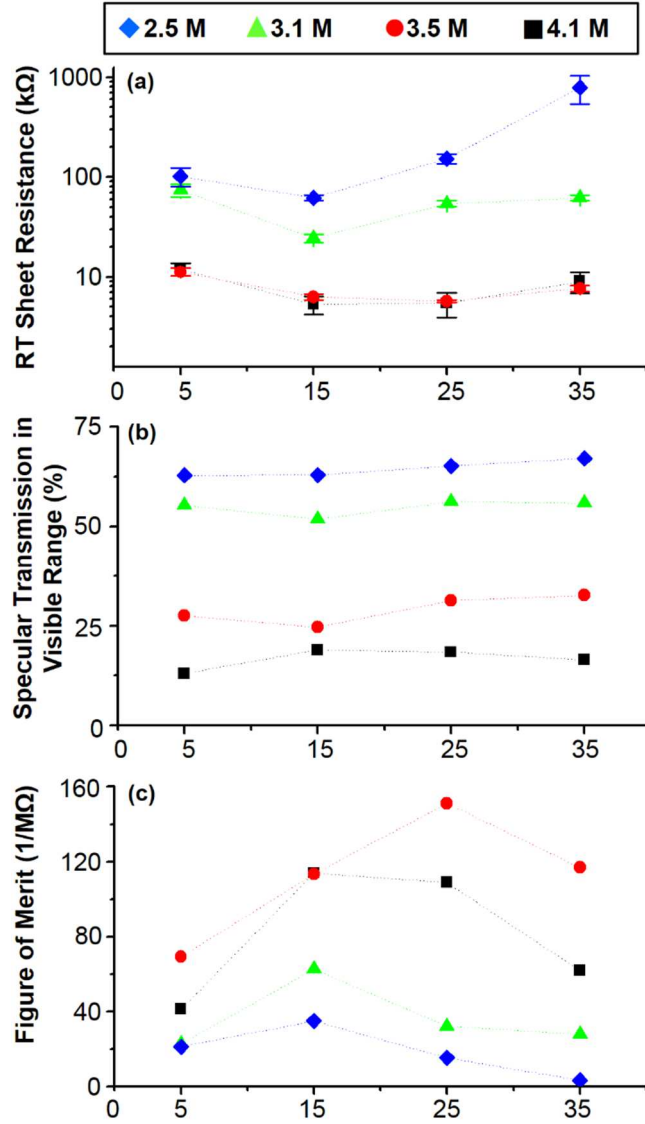


**Figure 4.1: TCO thin film of p-type  $\text{Ca}_3\text{Co}_4\text{O}_9$  obtained by spin coating and calcination of viscous resin on quartz substrate.** The FOM value for the  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin film is significantly higher than the FOM values of other solution processed p-type TCO thin films in literature and even higher than FOM values of most PVD or CVD prepared films. (a) Specular optical transmission of the TCO thin film between photon wavelengths of 250 nm and 3000 nm. The inset camera image at the top left shows the TCO thin film on a paper with printed logos. FOM and sheet resistance (SR in the figure) values for the TCO thin film are  $151 \text{ M}\Omega^{-1}$  and  $5.7 \text{ k}\Omega/\text{sq}$ , respectively. The inset graph at bottom right is in-plane Seebeck measurement of the

film, indicating a positive Seebeck coefficient of  $\sim 41 \mu\text{V/K}$  (p-type conductivity). **(b)** Cross-sectional SEM image of the  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin film in **(a)** showing film thickness of  $\sim 100 \text{ nm}$ . **(c)** Low (left) and high (right) magnification surface SEM images of the TCO thin film. The film is smooth and homogenous with nanometer scale voids and sparse micrometer scale cracks (left). The length of the nano-plates in the TCO film is  $\sim 50 - 100 \text{ nm}$  (right).

Decreasing the resin solute concentration to less than 3.5 M significantly increases sheet resistance and optical transmission of the calcined TCO thin films. In **Figure 4.2**, sheet resistances, average specular optical transmissions in the visible range, and FOM values are plotted against calcination times for all  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin films obtained from different solute concentrations (See **Figure C.2a**, Appendix C, for optical transmission spectrum of the 15 minute calcined films).<sup>25</sup> Sheet resistance values are reported with error bars (**Figure 4.2a**), which correspond to the standard deviation from the 12 sheet resistance measurements performed on each film. Both optical transmission and sheet resistance significantly increase with decreasing solute concentration for the TCO thin films that are prepared from 2.5, 3.1, and 3.5 M solute concentration resins (**Figure 4.2a**). This trend is most likely due to decreasing film thickness ( $t$ ) since both optical transmission and sheet resistance increase with decreasing  $t$  as expressed by  $T = e^{-\alpha t}$  and  $R_s = 1/(\sigma t)$  respectively (assuming fixed  $\alpha$  and  $\sigma$ ).<sup>27</sup> After 15 minutes calcination, the thicknesses of the TCO thin films that are prepared from 3.5, 3.1, and 2.5 M solute concentration resins are  $108 \pm 23$ ,  $85 \pm 9$ , and  $49 \pm 5 \text{ nm}$ , respectively, as measured by contact profilometry over a 1 mm length

with 1  $\mu\text{m}$  lateral sampling. This decrease in the film thickness with decreasing solute concentration is reasonable as low solute concentration resins have visibly lower viscosity, resulting in thinner films after spin coating and calcination.<sup>28,29</sup> Another reason for the increasing sheet resistance with decreasing solute concentration can also be associated with high porosity of the samples made with low solute concentration. Lower metal concentrations are expected to result in higher porosity as the water and organics are eliminated during baking (See **Figure C.3**, Appendix C, for surface SEM images of TCO films made from different solute concentration resins).<sup>25</sup> Decreasing the solute concentration from 3.5 to 3.1 M and from 3.1 to 2.5M causes up to a  $\sim 9.5\times$  and a  $\sim 12.8\times$  increase in the sheet resistance, respectively. The difference in sheet resistance between the 4.1 and 3.5 M samples is negligible (within the error limits) (**Figure 4.2a**). This negligible change in the sheet resistance for these two concentrations (4.1 and 3.5 M) is likely due to the low quality of the 4.1 M films in terms of homogeneity and uniformity. The coefficients of variation in the 12 sheet resistance measurements performed on 15 minute calcined thin films are 0.06, 0.09, 0.07 and 0.21 in order of increasing solute concentration. This reveals that 4.1 M sample is significantly more inhomogeneous compared to the others in terms of sheet resistance. The 4.1 M sample is also visibly inhomogeneous in terms of color and transparency, unlike the other films (See **Figure C.2b**, Appendix C, for camera images of the 15 minute calcined films).<sup>25</sup> SEM images indicate that the film prepared from 4.1 M solute concentration resin is non-uniform and not very well attached to the substrate (**Figure C.4**, Appendix C).<sup>25</sup>



**Figure 4.2: Sheet resistance, specular optical transmission and FOM values for  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin films obtained with different solute concentrations and calcination times.** Data points for the solute concentrations of 2.5, 3.1, 3.5, and 4.1 M are represented by blue diamond, green triangle, red circle, and black square symbols, respectively. (a) Average sheet resistance values obtained from 12 measurements on each TCO film vs. calcination time. The error bars are based on standard deviation in different sheet resistance values on each TCO film. (b) Average specular optical

transmission of the TCO thin films in visible range vs. calcination time. (c) FOM values calculated based on average sheet resistances and specular optical transmissions in visible range.

For all solute concentrations except 3.5 M, the lowest sheet resistances are observed at 15 minutes calcination and the sheet resistance gradually increases as calcination time increases past 15 minutes. For the 3.5 M solute concentration the sheet resistance gradually reaches the lowest value at 25 minutes calcination and increases again at 35 minutes calcination (**Figure 4.2a**). The high sheet resistance at 5 minutes calcination of the thin films is likely due to incomplete crystallization of the metallic species into the highly conductive  $\text{Ca}_3\text{Co}_4\text{O}_9$  phase. The increase in the sheet resistance as calcination times approach 35 minutes is likely due to coalescence of  $\text{Ca}_3\text{Co}_4\text{O}_9$  crystals, which break the electrical connections between the particles, as previously observed by Lee et. Al.<sup>30</sup> SEM images show obvious coalescence of  $\text{Ca}_3\text{Co}_4\text{O}_9$  crystals after 135 minutes of calcination in comparison with the TCO film calcined for 45 minutes, and a concomitant increase in sheet resistance by 2-3 orders of magnitude comparing the 45 and 135 minutes calcination time samples (**Figure C.5, Appendix C**).<sup>25</sup> The film obtained from the 3.5 M solute concentration resin at 25 minutes calcination has the lowest sheet resistance among all of our TCO thin films samples (Sheet resistance 5.7 k $\Omega$ /sq,  $\rho \approx 57$  m $\Omega$  cm). This same film showed the best spatial homogeneity in terms of sheet resistance, with a coefficient of variance of 0.03 in the 12 sheet resistance measurements performed on different locations on the film.

Optical transmission, interestingly, does not vary greatly across calcination



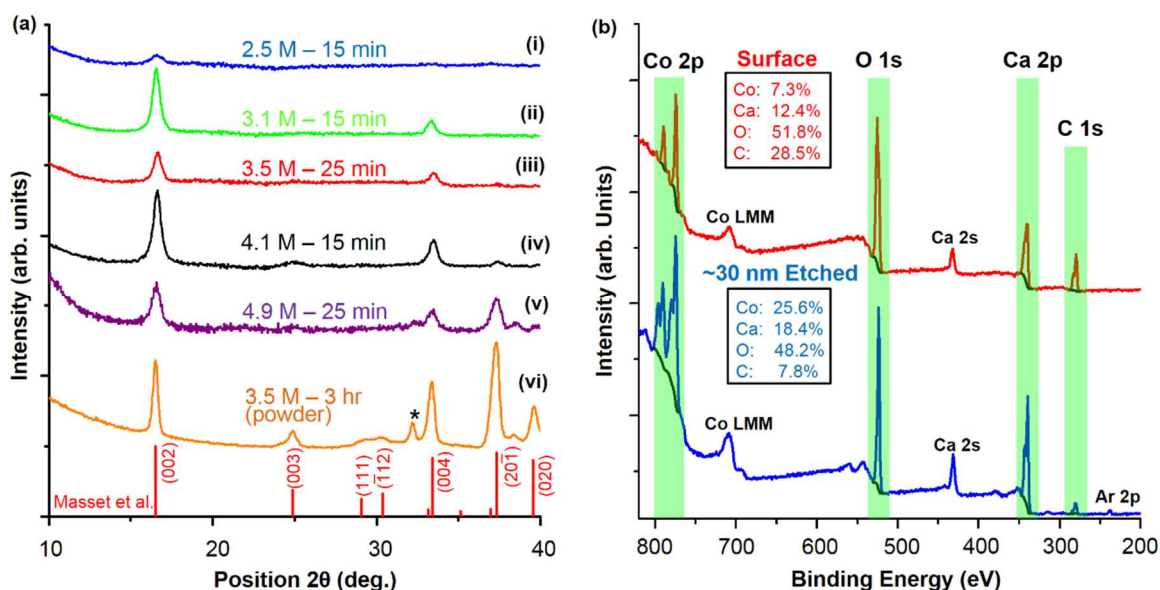
times (**Figure 4.2b**). However, transmission markedly increases as precursor concentration decreases. The highest visible range optical transmission (67% in average) is measured for the film obtained from the 2.5 M solute concentration after 35 minutes calcination.

The optimum transparent conductive properties are obtained when the baked films are calcined for 15 minutes (2.5, 3.1, and 4.1 M concentration) or 25 minutes (3.5 M). FOM values calculated from the sheet resistances and optical transmissions (**Figure 4.2a & 4.2b**) are plotted in **Figure 4.2c**. The FOM increases as the calcination time is increased from 5 minutes to 15 minutes for the 2.5, 3.1, and 4.1 M samples, and decreases as the calcination time increases further. For the 3.5 M sample the FOM peaks at 25 minutes calcination. The FOM increases as the solute concentration increases at all calcination times for solute concentrations 2.5, 3.1, and 3.5 M, but the highest solute concentration (4.1 M) shows a decreased FOM below the 3.5 M sample's values. The FOM is highest for the film with lowest sheet resistance (3.5 M solute concentration and 25 minutes calcination) and lowest for the film with highest optical transmission (2.5 M solute concentration and 35 minutes calcination).

XRD patterns obtained from the TCO thin films can be indexed by assuming  $\text{Ca}_3\text{Co}_4\text{O}_9$  crystalline structure. **Figure 4.3a** shows XRD patterns for four top-performing TCO thin films (solute resins concentrations: 2.5, 3.1, 3.5, and 4.1 M; calcined at 650 °C for 15 or 25 minutes) and another TCO thin film that is prepared from 4.9 M solute concentration resin and calcined at 650 °C for 25 minutes (**Figure 4.3a(i-v)**). For comparison, **Figure 4.3a** also includes the XRD pattern of calcium cobalt oxide powder that are obtained by direct calcination of the 3.5 M solute

concentration resin at 650 °C for 3 hours (**Figure 4.3a(vi)**). The resin solute concentrations and calcination times are indicated next to the XRD patterns. In XRD measurements of the films the scattering vector (Q) is aligned perpendicular to the quartz substrate. The wide background peak from the quartz substrate is separately measured and subtracted from all of the TCO thin film XRD patterns. The XRD pattern for the powder obtained from the calcination of the 3.5 M solute concentration resin (**Figure 4.3a(vi)**) matches with most of the reference peaks (**Figure 4.3a** red vertical bars, values taken from Masset et al.<sup>20</sup>) for the  $\text{Ca}_3\text{Co}_4\text{O}_9$  phase. The asterisk(\*) marked peak of the powder matches the (111) peak of CaO phase (PDF #004-0777), which can form as an impurity due to excess Ca in the resin. The XRD pattern of the TCO films prepared from a 4.1 and 4.9 M solute concentration resin have (002), (004) and  $(20\bar{1})$  peaks matching with the  $\text{Ca}_3\text{Co}_4\text{O}_9$  reference (**Figure 4.3a(iv-v)**). The XRD patterns for the other TCO thin films in **Figure 4.3a** clearly show two (00L) peaks from the  $\text{Ca}_3\text{Co}_4\text{O}_9$  reference (**Figure 4.3a(i-iii)**). The absence of peaks other than the (00L) planes indicates texturing of the  $\text{Ca}_3\text{Co}_4\text{O}_9$  crystals, with alignment of the c-axis perpendicular to the substrate. Considering the number of peaks from different crystal planes, the XRD patterns from the 4.1 and 4.9 M samples are a transition between the randomly oriented crystals of the powder form and the c-axis oriented crystals of the top-performing TCO thin films. This is probably due to extreme non-uniformity and excessive peeling of the TCO films prepared from the more viscous 4.1 and 4.9 M solute concentration resins. The peak from the CaO phase does not appear in the TCO thin film XRD patterns probably due to diffusion of excess Ca in the resin to the glass substrate. Scherrer analysis of the (002) peak from

the film with the highest FOM (3.5 M solute concentration and 25 minutes calcination, **Figure 4.3a(iii)**) indicates that the average crystal size is  $\sim 22$  nm along the  $[00L]$  direction. The film obtained from the 2.5 M solute concentration resin at 15 minutes calcination only shows a weak (002) peak that matches with the  $\text{Ca}_3\text{Co}_4\text{O}_9$  phase (**Figure 4.3a(i)**).



**Figure 4.3: XRD and XPS of TCO thin films.** (a) XRD for six  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin film samples (i-v) and  $\text{Ca}_3\text{Co}_4\text{O}_9$  powder (vi). The reference peak positions for  $\text{Ca}_3\text{Co}_4\text{O}_9$  phase from Masset et al. are shown with vertical red lines. The peak in powder sample (vi) denoted by asterisk (\*) is likely to originate from CaO impurity. (b) XPS of the top-performing TCO thin film (3.5 M solute conc., 25 min calc.). The Co 2p, O 1s, Ca 2p, and C 1s peaks are evaluated for elemental analysis (indicated by green bands). XPS is taken on the film surface (red curve) and after etching  $\sim 30$  nm into the film (blue curve). Subtracted backgrounds are shown with black curves. Atomic percentages are listed in the inset tables.

XPS was performed on the top-performing TCO thin film (3.5 M concentration, 25 minutes calcination) (**Figure 4.3b**). XPS was measured at the film surface (**Figure 4.3b, red curve**) and after removing the upper ~30 nm of surface by Ar-ion milling (**Figure 4.3b, blue curve**). The Co 2p, Ca 2p, O 1s and C 1s peaks are evaluated for elemental analysis. Elemental analysis from XPS shows that the film surface consists of 7.3% Co, 12.4% Ca, 51.8% O and 28.5% C. The Ca:Co ratio on the film surface is 1.7x higher than the original Ca:Co ratio in the resin, indicating Ca migration to the film surface. High C and Ca content on the film surface is likely from calcium carbonate or bicarbonate phases forming on the film surface as a result of the reaction of Ca ions in the film with H<sub>2</sub>O and CO<sub>2</sub> in air. After ~30 nm of Ar-ion milling, XPS indicates atomic percentages of 25.6 % Co, 18.4% Ca, 48.2% O and 7.8% C. This result implies that the Ca:Co ratio in the inner regions of the film is ~0.72, which is very close to the Ca:Co ratio of the stoichiometric Ca<sub>3</sub>Co<sub>4</sub>O<sub>9</sub> phase. However, the film is O deficient compared to the Ca<sub>3</sub>Co<sub>4</sub>O<sub>9</sub> phase, likely due to preferential etch of the O atoms by Ar-ion milling, as previously observed for other metal oxides in the literature.<sup>31-33</sup> The XPS after Ar-ion milling also shows a weak Ar 2p peak due to minor Ar deposition on the film (**Figure 4.3b**).

#### **4.4. Conclusion**

In conclusion, p-type transparent conducting thin films of Ca<sub>3</sub>Co<sub>4</sub>O<sub>9</sub> are obtained using a scalable and cost-effective manufacturing technique. The FOM value for the p-type Ca<sub>3</sub>Co<sub>4</sub>O<sub>9</sub> films can be as high as 151 MΩ<sup>-1</sup>, exceeding the FOM values reported for all other solution processed p-type TCO thin films and most of those prepared by PVD and CVD. The lowest sheet resistance and highest visible range

optical transmission for the TCO films are 5.7 k $\Omega$ /sq and 67.1%, respectively. Near infrared region optical transmission is as high as 85% for our most conductive TCO thin film. Our cost-effective, sol-gel based technique is compatible with most of the metallic species and it can be readily applied to other metal oxide thin film structures.

## REFERENCES

- <sup>1</sup> M. Aksit, S. K. Kolli, I. M. Slauch, and R. D. Robinson, *Appl. Phys. Lett.* (In Press) (2014).
- <sup>2</sup> D. S. Hecht, L. B. Hu, and G. Irvin, *Adv. Mater.* **23** (13), 1482 (2011).
- <sup>3</sup> D. Ginley, H. Hosono, and D.C. Paine, *Handbook of Transparent Conductors*. (Springer, 2010).
- <sup>4</sup> S. Sheng, G. J. Fang, C. Li, S. Xu, and X. Z. Zhao, *Phys. Status Solidi A- Appl. Mat.* **203** (8), 1891 (2006).
- <sup>5</sup> G. Hautier, A. Miglio, G. Ceder, G. M. Rignanese, and X. Gonze, *Nat. Commun.* **4** (2013).
- <sup>6</sup> E. Fortunato, P. Barquinha, and R. Martins, *Adv. Mater.* **24** (22), 2945 (2012).
- <sup>7</sup> H. Y. Liu, V. Avrutin, N. Izyumskaya, U. Ozgur, and H. Morkoc, *Superlattices Microstruct.* **48** (5), 458 (2010).
- <sup>8</sup> W. Beyer, J. Hupkes, and H. Stiebig, *Thin Solid Films* **516** (2-4), 147 (2007).
- <sup>9</sup> A. N. Banerjee and K. K. Chattopadhyay, *Prog. Cryst. Growth. Ch.* **50** (1-3), 52 (2005).
- <sup>10</sup> C. F. Windisch, K. F. Ferris, and G. J. Exarhos, *J. Vac. Sci. Technol. A.* **19** (4), 1647 (2001).

- 11 H. Kawazoe, M. Yasukawa, H. Hyodo, M. Kurita, H. Yanagi, and H. Hosono, Nature **389** (6654), 939 (1997).
- 12 G. Thomas, Nature **389** (6654), 907 (1997).
- 13 Y. Wang and H. Gong, Chem. Vapor. Depos. **6** (6), 285 (2000).
- 14 N. Duan, A. W. Sleight, M. K. Jayaraj, and J. Tate, Appl. Phys. Lett. **77** (9), 1325 (2000).
- 15 J. Tate, M. K. Jayaraj, A. D. Draeseke, T. Ulbrich, A. W. Sleight, K. A. Vanaja, R. Nagarajan, J. F. Wager, and R. L. Hoffman, Thin Solid Films **411** (1), 119 (2002).
- 16 R. Nagarajan, A. D. Draeseke, A. W. Sleight, and J. Tate, J. Appl. Phys. **89** (12), 8022 (2001).
- 17 E. Arca, K. Fleischer, and I. V. Shvets, Appl. Phys. Lett. **99** (11) (2011).
- 18 N. Uekawa and K. Kaneko, J. Phys. Chem. **100** (10), 4193 (1996).
- 19 S. H. Lim, S. Desu, and A. C. Rastogi, J. Phys. Chem. Solids **69** (8), 2047 (2008).
- 20 A. C. Masset, C. Michel, A. Maignan, M. Hervieu, O. Toulemonde, F. Studer, B. Raveau, and J. Hejtmanek, Phys. Rev. B **62** (1), 166 (2000).
- 21 M. Shikano and R. Funahashi, Appl. Phys. Lett. **82** (12), 1851 (2003).
- 22 M. Aksit, D. P. Toledo, and R. D. Robinson, J. Mater. Chem. **22** (13), 5936 (2012).
- 23 M. Aksit, B. C. Hoselton, H. J. Kim, D. H. Ha, and R. D. Robinson, ACS Appl. Mater. Interfaces **5** (18), 8998 (2013).
- 24 S. Dubinsky, G. S. Grader, G. E. Shter, and M. S. Silverstein, Polym. Degrad.

Stabil. **86** (1), 171 (2004).

- <sup>25</sup> See supplemental material at [URL will be inserted by AIP] for further characterization details regarding optical transmission, surface structure and sheet resistance of the TCO thin films. .
- <sup>26</sup> M. Suárez, A. Fernández, R. Torrecillas, and J. L. Menéndez, in *Sintering of Ceramics - New Emerging Techniques*, edited by A. Lakshmanan (InTech, 2012), pp. 542.
- <sup>27</sup> G. J. Exarhos and X. D. Zhou, *Thin Solid Films* **515** (18), 7025 (2007).
- <sup>28</sup> A. Aegerter, *Sol-Gel Technologies for Glass Producers and Users*. (Springer, 2004).
- <sup>29</sup> A. Busnaina, *Nanomanufacturing Handbook*. (Taylor & Francis, 2010).
- <sup>30</sup> J. Y. Lee, S. T. Connor, Y. Cui, and P. Peumans, *Nano Lett.* **8** (2), 689 (2008).
- <sup>31</sup> A. Kalabukhov, R. Gunnarsson, J. Borjesson, E. Olsson, T. Claeson, and D. Winkler, *Phys. Rev. B* **75** (12) (2007).
- <sup>32</sup> D. S. Kan, T. Terashima, R. Kanda, A. Masuno, K. Tanaka, S. C. Chu, H. Kan, A. Ishizumi, Y. Kanemitsu, Y. Shimakawa, and M. Takano, *Nat. Mater.* **4** (11), 816 (2005).
- <sup>33</sup> C. Morant, J. M. Sanz, and L. Galan, *Phys. Rev. B* **45** (3), 1391 (1992).

## CHAPTER 5

### 5. DIRECT MEASUREMENTS OF SURFACE SCATTERING IN Si NANOSHEETS USING A MICROSCALE PHONON SPECTROMETER: IMPLICATIONS FOR CASIMIR-LIMIT PREDICTED BY ZIMAN THEORY\*

#### **5.0. Abstract:**

Thermal transport in nanostructures is strongly affected by phonon-surface interactions, which are expected to depend on the phonon's wavelength and the surface roughness. Here we fabricate silicon nanosheets, measure their surface roughness ( $\sim 1$  nm) using AFM, and assess the phonon scattering rate in the sheets with a novel technique: a microscale phonon spectrometer. The spectrometer employs STJs to produce and detect controllable non-thermal distributions of phonons from  $\sim 90$  to  $\sim 870$  GHz. This technique offers spectral resolution nearly 10 times better than a thermal conductance measurement. We compare measured phonon transmission rates to rates predicted by a MC model of phonon trajectories, assuming that these trajectories are dominated by phonon-surface interactions and using the Ziman theory to predict phonon-surface scattering rates based on surface topology. Whereas this model predicts a diffuse surface scattering probability of less than 40%, our measurements are consistent with a 100% probability. Our nanosheets therefore exhibit the so-called 'Casimir limit' at a much lower frequency than expected if the

---

\* Originally Published as: Jared B. Hertzberg <sup>†</sup>, Mahmut Aksit <sup>†</sup>, Obafemi O. Otelaja <sup>†</sup>, Derek A. Stewart, and Richard D. Robinson, " Direct Measurements of Surface Scattering in Si Nanosheets using a Microscale Phonon Spectrometer: Implications for the Casimir-Limit Predicted by Ziman Theory", *Nanoletters* 14 (2), 403-415 (2014). <sup>†</sup> Equally contributing authors. Reprinted with Permission from American Chemical Society.



phonon scattering rates follow the Ziman theory for a 1 nm surface roughness. Such a result holds implications for thermal management in nanoscale electronics and the design of nanostructured thermoelectrics.<sup>1</sup>

### 5.1. Introduction

Developing experimental tools to understand heat flow at nanoscale dimensions is a grand challenge of nanoscience.<sup>2-9</sup> While recent works have demonstrated the ability to accurately predict thermal conductivity in bulk materials;<sup>10-12</sup> phonons – the primary heat carrier in dielectrics – are expected to behave differently in nano-dimensional channels and structures.<sup>13-18</sup> Our lack of experimental diagnostics in this area has created bottlenecks to understanding the basic physics of phonons. Unresolved questions include frequency-dependent phonon dynamics and transport through nanostructures, the effects of acoustic confinement on transport, and the frequency-dependence of phonons scattered by boundaries.

The limit on thermal conductivity through finite-sized channels has been historically described by the “Casimir limit.” In the classical kinetic model, thermal conductivity can be expressed as  $\kappa_{ph} = \frac{1}{3} v_{ph} C_V \Lambda$ , where  $v_{ph}$  is the group velocity of the phonons,  $C_V$  is the specific heat, and  $\Lambda$  is the phonon mean free path.<sup>19</sup> In the boundary-scattering regime, in which phonon-surface scattering dominates all other phonon scattering mechanisms, the mean free path  $\Lambda$  will equal the surface-scattering mean free path  $l_{eff}$ . In the limit of extremely roughened surfaces the phonons will scatter diffusively from the walls and  $l_{eff}$  reduces to the ‘Casimir limit’ mean free path  $l_0$ , which is a function of both the height and width of the channel.<sup>20</sup> For over seventy years, efforts to include the effects of surface roughness on both electronic

and thermal transport have focused primarily on the concept of a specularity parameter,  $p$ , first introduced by Fuchs in 1938 in modeling electrical transport in thin films.<sup>21</sup> In the Boltzmann transport equation,  $p$  is employed as a phenomenological parameter to set boundary conditions on the solutions at the channel surfaces. A value  $p = 0$  corresponds to a surface that diffusively scatters the phonons 100% of the time, while  $p = 1$  corresponds to a perfectly specular surface (see also Appendix D.0 for further discussion).<sup>4,22,23</sup> Borrowing from diffraction theory, Ziman offered a physical basis for the specularity by expressing it as a function of the phonon wavelength  $\lambda$  and a surface roughness or ‘asperity parameter’  $\eta$ , defined as the standard deviation of the local surface amplitude. In Ziman’s theory, the mean free path  $l_{eff}$  of a phonon of wavelength  $\lambda$  in a channel of characteristic dimension  $l_0$  and roughness  $\eta$  may be described:<sup>4,22,24-26</sup> <sup>1</sup>

$$p(\lambda) = e^{-(4\pi\eta/\lambda)^2} \quad \text{Eq. 5.1}$$

$$l_{eff} = l_0 \frac{1+p(\lambda)}{1-p(\lambda)} \quad \text{Eq. 5.2}$$

While the Ziman specularity parameter has been widely used to explain experimental measurements of phonon boundary scattering, the results are often

---

<sup>1</sup> The specularity expression presented by Ziman<sup>21</sup> includes an additional power of  $\pi$  not found in later analyses.<sup>23-25</sup> We follow the later work and omit the power of  $\pi$ , which appears to be an error in the original work. We thank Alexei Maznev for bringing this issue to our attention.

contradictory or inconclusive. The mean free paths of acoustic phonons in ultra-thin silicon membranes measured with optical pump probe techniques<sup>27</sup> and the thermal conductance of silicon nanowires in the mesoscopic size limit<sup>4</sup> have been shown to match the Ziman expression. However, in cases where the model has been applied to thermal conductance measurements of suspended membranes<sup>28</sup> and nanocrystalline silicon<sup>29</sup>, as well as radiative ballistic phonon transport in suspended membranes<sup>30,31</sup>, the Ziman expression alone does not explain the experimental values for mean free paths or specularly parameters. In addition, models that fit the specularly parameter to the measured thermal conductivity of Si nanowires often appear to require diffusive scattering beyond the Casimir limit<sup>32,33</sup>, or else permit multiple specularly values to fit the same data set.<sup>34,35</sup> It remains unclear whether such disparities stem from unexplored experimental factors, or from limitations of the Ziman model. In particular, correlating thermal transport measurements to measured surface roughness at the nanoscale has not been widely attempted. To our knowledge, only one such study appears in the recent literature, employing single crystalline Si nanowires and a limited TEM projection method to assess surface roughness.<sup>8</sup> The remaining open questions motivate a more direct method for examining the validity of the model.

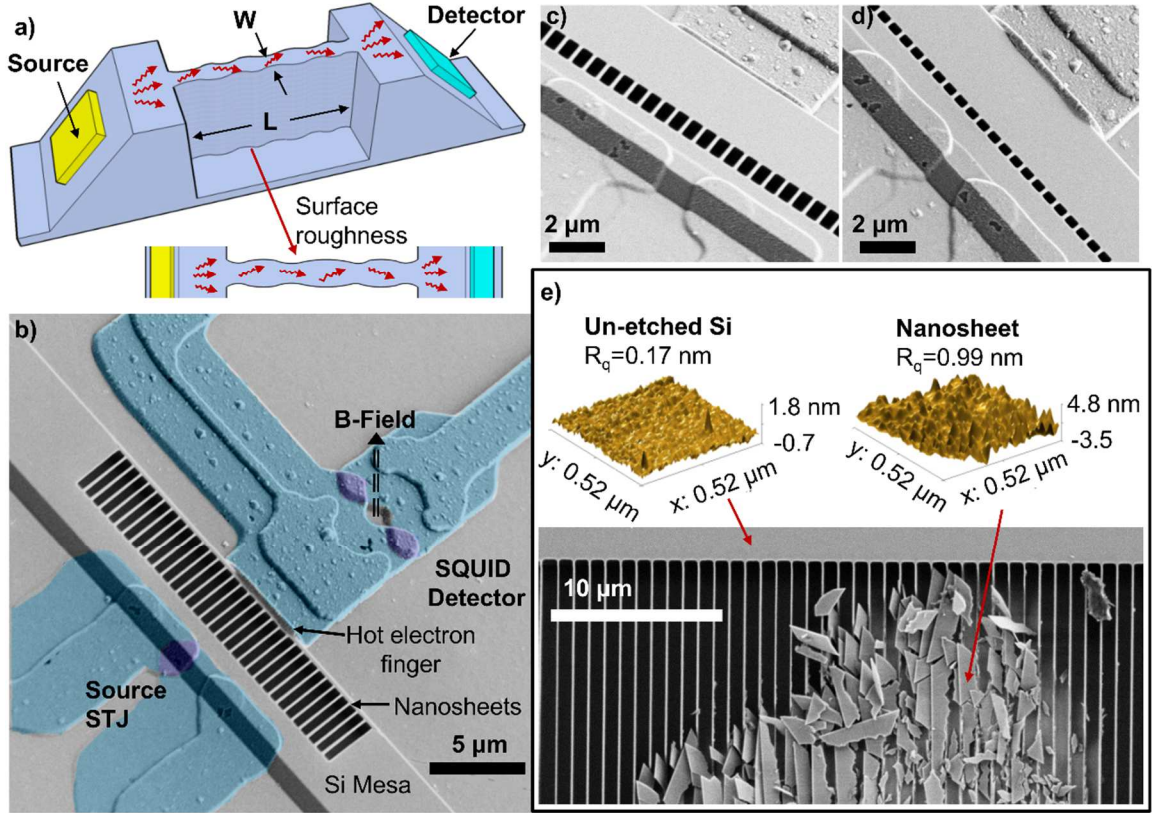
Existing experimental studies of the Casimir-Ziman theory rely on measurements of thermal conductivity  $\kappa_{ph}$  to probe the wavelength dependence of  $l_{eff}$ .<sup>4,7,36,37</sup> Thermal phonons comprise a Planck distribution of frequencies. In the Debye model, at low temperatures phonons in the interval  $d\omega$  around frequency  $\omega$  contribute to the specific heat  $C_V$  and thereby to  $\kappa_{ph}$  a fraction proportional to

$e^{\frac{\hbar\omega}{k_B T}} \cdot \left(\frac{\hbar\omega}{k_B T}\right)^4 d\omega / \left(e^{\frac{\hbar\omega}{k_B T}} - 1\right)^2$ . This distribution is peaked at the so-called 'dominant phonon frequency'  $\omega_{dom} = 3.83k_B T / \hbar$  but is quite broad, and this naturally limits the spectral precision in assessing  $l_{eff}$ .<sup>4,36-38</sup> For instance, to measure scattering by phonons in the interval 390-410 GHz, we may measure  $\kappa_{ph}$  at  $T = 5.0$  K, where  $\omega_{dom} = 2\pi \cdot 400$  GHz. However, the phonons in the 20 GHz interval around the peak contribute only 3.6% of  $\kappa_{ph}$ . The measured value of  $l_{eff}$  will thus represent a much larger range of phonon frequencies.

Some recent studies have attempted to address this spectral imprecision by varying heating areas in order to distinguish different mean-free-path regimes.<sup>39-41</sup> A technique using a narrow distribution of phonons, on the other hand, would allow more precise assessment of the Casimir limit. Such narrow-band phonon sources are realizable using superlattice structures optically-pumped to produce acoustic emission<sup>42</sup>, but can require complicated engineering, especially to couple the phonons into a nanostructure such as a nanowire or nanosheet.

In this work, we utilize an alternative narrow-band phonon probe that is compact, simply fabricated, scalable, and excellently suited to studying scattering behavior in nanostructures: a microscale phonon spectrometer employing thin-film aluminum STJs as phonon source and detector (**Figure 5.1a, 5.1b**). Rather than measuring thermal conductance, this device measures the phonon power transmitted through nanostructures as a function of phonon frequency. It employs a non-thermal distribution of phonons, in which a large fraction of the phonon power lies in a band  $\sim 20$  GHz around the peak.<sup>43</sup> For peak frequencies of several hundred GHz, the ratio

of peak power to total power ( $P_{peak}/P_{tot}$ ) exceeds 30%, thus enabling a spectral precision nearly 10 times that of a thermal measurement. We apply this technique to observe phonon transmission through arrays of silicon nanosheets (**Figure 5.1b-d** and **Table 5.1**), and compare our results to theory by incorporating careful measurements of the nanosheet surface roughness (**Figure 5.1e**). Nanostructured silicon is of interest because of silicon's critical importance to semiconductor technology and its potential application as an efficient thermoelectric<sup>2,44</sup>, and nanosheet geometries have also received interest for thermoelectric applications.<sup>45-47</sup> Our precise measurement of surface roughness combined with precise control of phonon frequency allows for a unique characterization of the contributions of phonon surface scattering to the total phonon transmission, and this, in turn, has implications for predictions made by the Casimir-Ziman theory.



**Figure 5.1: Design of Spectrometer to Probe Phonon Transmission through**

**Silicon Nanosheet Arrays.** **a)** Schematic of phonon transmission measurement geometry. Source transducer emits phonons of known frequency, which travel ballistically in line-of-sight through bulk silicon crystal, enter the nanosheet, scatter at the nanosheet surfaces, emerge from far end and are collected by a detector. **b)** False-colored SEM image of spectrometer containing nanosheet array type C (3.9  $\mu\text{m}$  long nanosheets, see **Table 5.1**). Nanosheets are embedded into 0.8  $\mu\text{m}$  high by 7  $\mu\text{m}$  wide silicon mesa, formed on top of a 525  $\mu\text{m}$  thick Si chip. The full set of measured nanosheets (see **Table 5.1**) has dimensions ranging from 0.2 to 3.9  $\mu\text{m}$  long, 0.12 to 0.38  $\mu\text{m}$  wide and 0.6 to 0.8  $\mu\text{m}$  high. Source STJ comprises two Al layers of thickness 20 nm and 79 nm on side-wall of silicon mesa. Detector comprises a 'hot

electron finger' in contact with opposite side-wall of the mesa, which intercepts the phonon flux, and a double-junction STJ to register the resulting tunnel current. **c)** SEM image of a spectrometer with 1  $\mu\text{m}$  long nanosheets (similar to type B, see **Table 5.1**). **d)** SEM image of spectrometer with nanosheet array type A (0.4  $\mu\text{m}$  long nanosheets, see **Table 5.1**). **e)** (top) Atomic-force microscope measurement of surface roughness of typical un-etched Si wafer surface (top left), and Si nanosheet sidewall surface (top right) for comparison. Vertical scales are -0.7 to 1.8 nm (top left) and -3.5 to 4.8 nm (top right). Standard deviation  $R_q$  of surface amplitude is equivalent to roughness  $\eta$ . (bottom) SEM image of etched nanosheets showing method for breaking nanosheets to permit AFM measurements of their surface roughness.

Nano-sheet type	Array Pitch ( $\mu\text{m}$ )	Sheet Length, $L$ ( $\mu\text{m}$ )	Sheet Width, $W$ ( $\mu\text{m}$ )	Sheet Height ( $\mu\text{m}$ )	Source STJ Width ( $\mu\text{m}$ )	Detector Finger Width ( $\mu\text{m}$ )	Detector Finger Height ( $\mu\text{m}$ )	Simulated Nanosheet Transmission Factors		Measured phonon transmission signal $\delta I_d / \delta I_s$ , peak frequency 400 GHz ( $\times 10^3$ )
								$[ \Gamma_{los}(s \rightarrow ns \rightarrow d) + \Gamma_{sc}(s \rightarrow ns \rightarrow d) + \Gamma_{los}(s \rightarrow d) ] \times 10^3$	Using specularity = 0 Using specularity found from Eq. (1) for $\eta = 1 \text{ nm}$ , $f_{\text{peak}} = 400 \text{ GHz}$ and $P_{\text{peak}}/P_{\text{tot}} = 0.32$	
A	0.60	0.40	0.15	0.75	2.11	5.38	0.80	3.67	6.35	4.02 +/- 0.27
B	0.60	1.01	0.15	0.80	2.17	5.53	0.80	1.77	4.93	2.08 +/- 0.27
C	0.60	3.94	0.14	0.80	1.68	6.12	0.80	0.29	2.15	0.28 +/- 0.13
D	0.30	0.21	0.13	0.60	2.35	6.22	0.55	7.25	10.01	10.8 +/- 0.3
E	0.30	0.59	0.12	0.65	2.08	6.27	0.54	4.14	7.13	6.41 +/- 0.31
F	0.30	0.99	0.13	0.65	2.08	6.28	0.53	3.45	6.50	3.16 +/- 0.26
G	0.30	2.99	0.13	0.70	2.15	1.79	0.47	0.61	1.94	1.02 +/- 0.32
H	0.65	0.57	0.33	0.70	1.46	2.75	0.80	5.53	8.28	9.36 +/- 0.60
I	0.65	0.94	0.34	0.70	1.37	5.84	0.80	6.96	13.41	8.70 +/- 0.60
J	0.65	3.94	0.38	0.80	1.93	1.88	0.53	1.12	4.75	1.70 +/- 0.14

**Table 5.1: Geometry of phonon spectrometers and measured silicon nanosheets.**

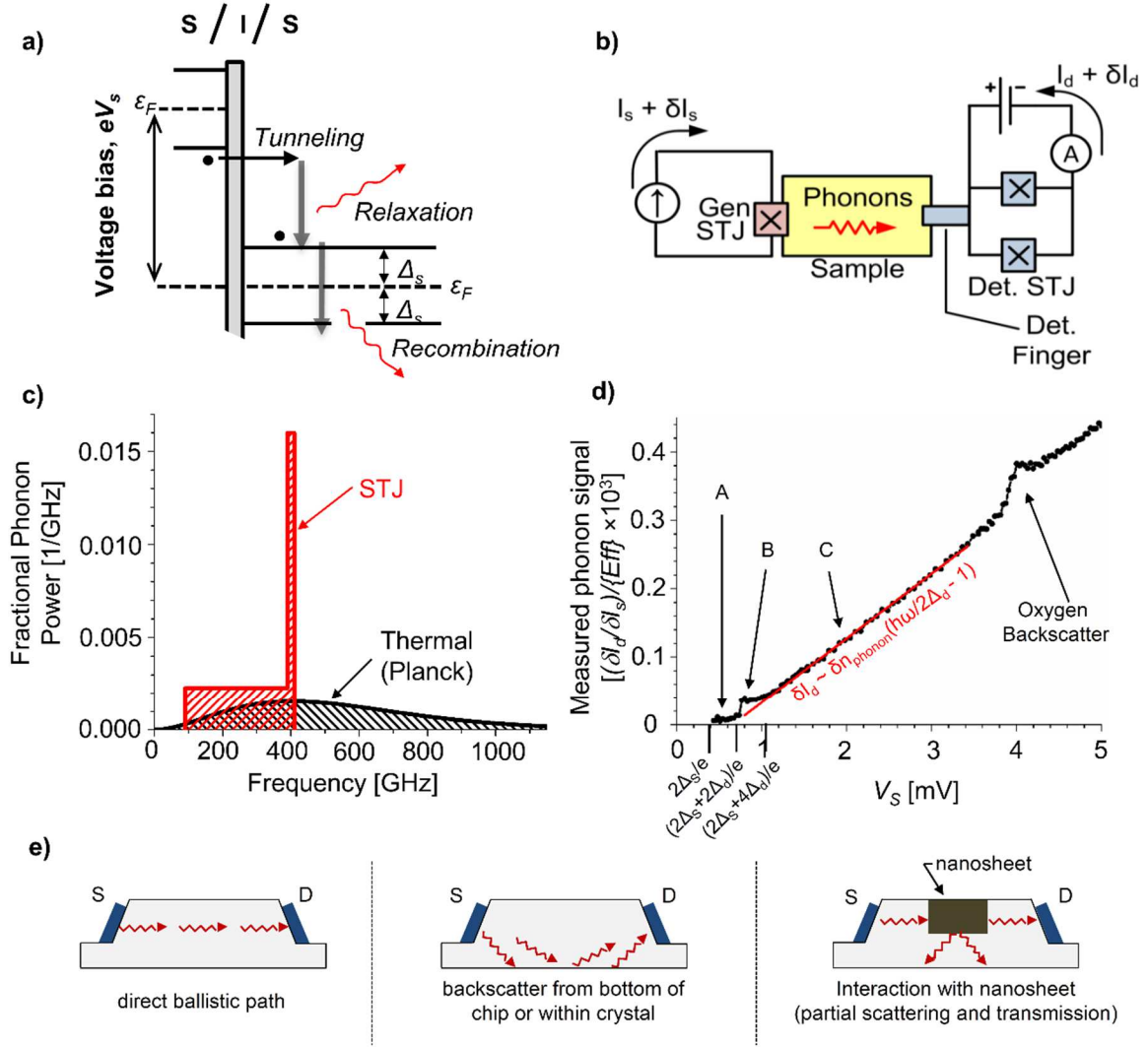
Each lettered spectrometer comprises an STJ phonon source, nanosheet array, and STJ phonon detector fabricated on a 7  $\mu\text{m}$  wide by 0.8  $\mu\text{m}$  high mesa. All dimensions are

in  $\mu\text{m}$ . Calculated specularly assumes roughness  $\eta = 1 \text{ nm}$ .

## **5.2. Results and Discussion**

Our measurement and principles of operation are illustrated in **Figure 5.1a**, **5.1b**, **5.2a**, and **5.2b**. A non-thermal phonon flux emerging ballistically from an STJ source transducer enters the nanosheet; the portion emerging at the far end is collected by the detector. We isolate narrow portions of phonon spectrum by modulating the source STJ's current ( $I_s$ ) at a frequency of a few Hz and monitoring the resulting modulations in the detector current  $I_d$ . The measured differential transfer function  $\delta I_d / \delta I_s$  tells us the fraction of the phonon power within the chosen spectrum that is transmitted to the detector.<sup>43,48-50</sup>





**Figure 5.2: Phonon Spectrometer Operating Principles.** a) Band diagram of quasiparticle tunneling and phonon decay processes in source STJ of superconducting gap  $\Delta_s$ , biased at voltage  $V_s \geq 2\Delta_s/e$ . Electrons tunnel from superconducting (S) layer through insulating (I) barrier, entering excited states in second superconducting (S) layer. As they decay (relax) from these states towards the edge of superconducting gap, they emit acoustic phonons. The resulting distribution of phonons exhibits a sharp step-edge at energy  $eV_s - 2\Delta_s$ . Further decay into the Cooper-paired state (recombination) emits additional phonons of energy  $\sim 2\Delta_s$ . b) Schematic of device

operation and wiring. Bias current  $I_s$  holds source STJ at fixed voltage  $V_s$  while source modulation  $\delta I_s$  selects differential portion of emitted spectrum. Resulting modulation  $\delta I_d$  in detector current indicates differential phonon power transmitted through the sample. **c)** Spectral density of differential phonon power, estimated for source STJ of **Figure 5.1b**, biased at  $\sim 2.1$  mV. Modulation  $\delta I_s$  produces differential spectrum with peak frequency  $(eV_s - 2\Delta_s)/h = 400\text{GHz}$  and width  $\delta\omega/2\pi = 20$  GHz. For comparison, we have also included the power spectral density of thermal phonons,  $\sim \frac{\hbar\omega^3 d\omega}{2\pi^2} / (e^{\hbar\omega/k_B T} - 1)$ , which at temperature 6.8 K exhibits peak power at  $\omega = 2\pi \cdot 400$  GHz. Each spectrum is normalized so that total detectable power (power carried by phonon frequencies  $\omega/2\pi > 90$  GHz) is unity. **d)** Phonon transmission through bulk silicon, typical measurement. Differential transfer function  $\delta I_d/\delta I_s$  indicates fraction of phonon power emitted from the source that arrives at the detector. Phonon emission & detection regimes A, B, C correspond to the three regimes of Eq. 5.3. Feature at  $V_s \sim 4$  mV indicates  $\sim 870$  GHz resonant backscatter from oxygen impurities in silicon substrate.<sup>51</sup> **e)** Phonon pathways between STJ source (S) and detector (D). Acoustic energy can either travel ballistically unimpeded in line-of-sight from source to detector (rates  $\dot{n}_{los}(s \rightarrow d)$  and  $\dot{n}_{los}(s \rightarrow ns \rightarrow d)$ ), scatter from bottom of silicon chip or from scattering centers within the chip ( $\dot{n}_{bs}$ ), or partially scatter within nanosheets placed between source and detector ( $\dot{n}_{sc}(s \rightarrow ns \rightarrow d)$ ).

To produce narrow-band phonons, we exploit emission processes in STJs at a temperature  $T \cong 0.3$  K.<sup>43,48-50,52-54</sup> Our phonon source comprises an aluminum STJ

having superconducting gap  $2\Delta_s \cong 400 \mu\text{eV}$ , biased at a voltage  $V_s > 2\Delta_s/e$  to break the electrons out of the paired state into quasiparticle excited states, and drive them across the tunnel barrier (**Figure 5.2a**). The tunneled electrons relax to lower energies by emission of acoustic phonons in a broad non-thermal energy distribution with a sharp cutoff at energy  $eV_s - 2\Delta_s$  (relaxation phonons in **Figure 5.2a**).<sup>50,54,55</sup> The phonons are incoherent and to a first approximation will have both random polarization and random direction due to elastic scattering of the tunneled electrons within the Al film.<sup>50</sup> Further decay by recombination into Cooper pairs releases additional phonons of fixed energy  $\sim 2\Delta_s$ , which at bias voltages  $V_s \gg 4\Delta_s/e$  contribute a small additional emitted power (recombination phonons in **Figure 5.2a**).<sup>50</sup>

For a STJ of normal-state tunnel resistance  $R_n$  (**Table 5.2**), the small AC modulation  $\delta I_s$  adds a modulation  $\delta V_s = \delta I_s R_n$  to voltage  $V_s$ . The resulting modulation  $2V_s \delta I_s$  in the power emitted by the STJ comprises a differential distribution of phonons that is sharply peaked at the spectral cutoff energy  $\hbar\omega_{peak} = eV_s - 2\Delta_s$ , as illustrated in **Figure 5.2c**.<sup>48-50,54</sup> Taking the total differential phonon power  $P_{tot} \approx 2V_s \delta I_s$  to be the integral  $\int \hbar\omega \dot{n}_{ph,s}(\omega) d\omega$  of power across the phonon distribution, a spectrum having  $P_{peak} = P_{tot}$  would comprise phonons of frequency  $\omega_{peak}$  emitted at rate  $\dot{n}_{ph,s} = 2\delta I_s/e$ .<sup>43</sup>

Nanosheet Type	Source STJ $R_n$ ( $\Omega$ )	Det. STJ $R_n$ ( $\Omega$ )	$2\Delta_s$ ( $\mu\text{eV}$ )	$2\Delta_d$ ( $\mu\text{eV}$ )	Source IV Step Width ( $\mu\text{eV}$ )	Modul. Amp $\delta V_s$ ( $\mu\text{V rms}$ )	Max. Detector Linearity Range (GHz)	Det. Noise ( $\text{fA}/\sqrt{\text{Hz}}$ )	Estimated Detector Efficiency $\{Eff\}$
A	700	128	414	363	76	7.37	643	110	0.11
B	733	123	418	356	81	7.37	696	140	0.11
C	935	167	419	367	58	9.82	491	90	0.08
D	762	116	415	353	71	9.84	619	55	0.12
E	786	113	411	353	71	9.87	696	118	0.12
F	790	115	409	365	72	9.06	649	130	0.12
G	726	117	405	351	66	10.32	698	80	0.12
H	1148	197	400	365	68	9.39	687	70	0.07
I	1221	211	403	368	70	8.80	687	120	0.07
J	895	138	394	367	54	8.00	693	60	0.10

**Table 5.2: Electronic characterization of the phonon spectrometers for silicon**

**nanosheet measurements.** Each lettered spectrometer comprises an STJ phonon source, nanosheets, and an STJ phonon detector. Letters (Nanosheet Type) correspond to same devices as listed in **Table 5.1**.

Phonons are detected using a second STJ. The detector is formed in a double-junction (SQUID) configuration so that its Josephson current may be suppressed by an applied magnetic field, and is biased at voltage  $\sim 180 \mu\text{V}$ , allowing precise measurement of its quasiparticle tunnel currents. A ‘finger’ of superconducting Al serves to collect phonons from a well-defined region, and the total detector current  $I_d$  rises in response to total incident phonon flux (**Figure 5.2b**). The modulated detector current  $\delta I_d$  resulting from the differential portion of incident phonons is distinguished using a lock-in amplifier.<sup>43</sup> Since the detector STJ is sensitive only to phonons of energy greater than the gap value  $2\Delta_d/h \approx 90 \text{ GHz}$ , the measured phonon population comprises phonons of frequencies between  $\sim 90 \text{ GHz}$  and  $(eV_s - 2\Delta_s)/h$ , with a sharp peak at frequency  $(eV_s - 2\Delta_s)/h$ . In practice, this spectrometer allows us to probe our nanosheets quasi-monochromatically over a range of frequencies from  $\sim 100$  to  $\sim 870$

GHz (wavelengths  $\sim 60$  to  $\sim 7$  nm in Si).<sup>43</sup> In addition to studies of nanosheet surface scattering, this range should allow these spectrometers to probe a wide variety of effects in silicon. For instance,  $\sim 800$  GHz phonons in Si have been reported to exhibit dispersion-related effects on the anisotropy of propagation.<sup>56</sup> The energy resolution of our measurement is  $\sim 60$ - $80$   $\mu\text{eV}$  ( $\sim 15$  to  $20$  GHz). Sensitivity is limited by detector noise level, the lowest achieved being  $\sim 60$  fA/ $\sqrt{\text{Hz}}$  at a modulation frequency of  $11$  Hz (**Table 5.2**). The corresponding noise equivalent power, NEP, for phonon detection is  $\sim 10^{-15}$  W/ $\sqrt{\text{Hz}}$ , or  $\sim 2 \times 10^7$  phonons of energy  $\sim 2\Delta_d$  per second per  $\sqrt{\text{Hz}}$ .<sup>43</sup>

The detector's response to incident phonons may be modeled by considering quasiparticle-phonon interactions (presented elsewhere<sup>43</sup>). If phonons of frequency  $\omega$  strike the finger at a differential rate  $\dot{n}_{ph,d}$ , then the average differential rate of quasiparticle production  $\dot{n}_{QP,ph}$  within the detector film should be

$$\begin{aligned} \dot{n}_{QP,ph} &= 0 & \text{for } \hbar\omega < 2\Delta_d \\ \dot{n}_{QP,ph} &= T_{SiAl} \cdot \alpha_{abs}(\omega) \cdot 2\dot{n}_{ph,d}(\omega) & \text{for } 2\Delta_d \leq \hbar\omega < 4\Delta_d \\ \dot{n}_{QP,ph} &= T_{SiAl} \cdot \alpha_{abs}(\omega) \cdot 2 \left( \frac{\hbar\omega}{2\Delta_d} - 1 \right) \dot{n}_{ph,d}(\omega) & \text{for } \hbar\omega \geq 4\Delta_d \end{aligned}$$

Eq. 5.3

In Eq. 5.3, factor  $T_{SiAl}$  is the acoustic transmission factor for phonons transiting from Si into Al. For incidence angles not far from normal, this factor should be  $>0.9$ .<sup>57</sup> The fraction of phonons  $\alpha_{abs}(\omega)$  absorbed in the detector finger will depend on the frequency-dependent phonon absorption lengths in Al, and is of order  $0.5$  for these phonon frequencies and the thickness of our detector film.<sup>43,58</sup> To calculate the total

differential rate  $\langle \dot{n}_{QP,ph} \rangle$  of quasiparticles produced by incident phonons, Eq. 5.3 must be integrated across the differential spectrum of incident phonons. The measurable differential detector tunnel current  $\delta I_d$  is then related to quasiparticle production rate via  $\delta I_d = e \cdot \{Eff\} \cdot \langle \dot{n}_{QP,ph} \rangle$ , where  $\{Eff\}$  is a non-dimensional efficiency factor that accounts for the rate of tunnel current compared to other quasiparticle loss processes in the aluminum.<sup>43</sup> In practice, the measured  $\delta I_d / \delta I_s$ , plotted as a ‘spectrum’ against phonon peak frequency, follows closely the three regimes represented in Eq. 5.3. This is illustrated in **Figure 5.2d**, where the regimes are indicated by letters A, B, C.

To enable consistent comparisons among different detectors’ signals, we scale each measured differential transfer function  $\delta I_d / \delta I_s$  by the value of  $\{Eff\}$  for that detector.<sup>43</sup> We calculate  $\{Eff\}$  for each detector using conventional theories of tunneling rate and quasiparticle recombination. This efficiency factor is inversely proportional to the detector’s tunnel-barrier resistance, which may be measured to allow calculation of  $\{Eff\}$ , resulting in typical values of  $\sim 0.1$  (**Table 5.2**).<sup>43</sup> For moderate increases in total detector current  $I_d$ , this efficiency remains constant and therefore detector response remains linear and Eq. 5.3 remains valid.<sup>43</sup> In **Table 5.2**, we report for each spectrometer the corresponding maximum phonon peak frequency, designated ‘Maximum detector linearity range’. Reported phonon transmission signals are restricted to this range.

The spectrometers are fabricated using photolithography, wet chemical etch and thin-film deposition.<sup>49</sup> We first form  $0.8 \mu\text{m}$  high silicon ‘mesas’ on top of a  $525 \mu\text{m}$  thick Si chip. The top surface of the mesa is a [100] plane of the Si crystal. (Full description of spectrometer geometry and fabrication are discussed elsewhere<sup>43</sup>). For

phonon scattering studies, we embed nanosheet arrays of desired geometry into the bulk silicon of the mesa. The entrance to each array is positioned  $1.9\ \mu\text{m}$  from the side-wall holding the STJ phonon source. The transmission through an array of identical nanosheets is larger and more easily measured than through a single sheet. We define the structures using electron-beam lithography and selectively etch them into the silicon using a non-scalloping DRIE plasma etch technique (**Figure 5.1b,c,d** and **Table 5.1**).

Our nanosheets have height  $0.6$  to  $0.8\ \mu\text{m}$  and width  $0.12\ \mu\text{m}$  to  $0.38\ \mu\text{m}$  (See **Table 5.1**). The nanosheets are anchored to the substrate, so that their bottom end offers an avenue for phonons to escape, while their top surface is smooth polished Si and therefore likely to reflect specularly. The plasma-etched side-walls, however, constitute the great majority of surface area and therefore we expect the roughness of these walls to dominate the phonon scattering in the channel. The Casimir-limit mean free path  $l_0$  should therefore be similar to the sheet width. At room temperature, such dimensions are comparable to published estimates of  $250\text{-}300\ \text{nm}$  for the mean free path of phonon thermal transport in silicon.<sup>19,59</sup> At temperatures below  $1\ \text{Kelvin}$ , however, phonon-phonon scattering is negligible.<sup>23,56,60</sup> Spontaneous decay of acoustic phonons, while possible even at very low temperatures, should also be negligible for frequencies below  $1\ \text{THz}$ , as shown previously by researchers.<sup>61</sup> Surface scattering should therefore dominate. We performed AFM measurements of etched nanosheet sidewall surfaces (**Figure 5.1e**). AFM offers a highly accurate two-dimensional measurement, in contrast to the line-edge roughness found by methods such as TEM.<sup>7,62</sup> The measurement region, several phonon wavelengths in extent, exhibits an approximately Gaussian distribution

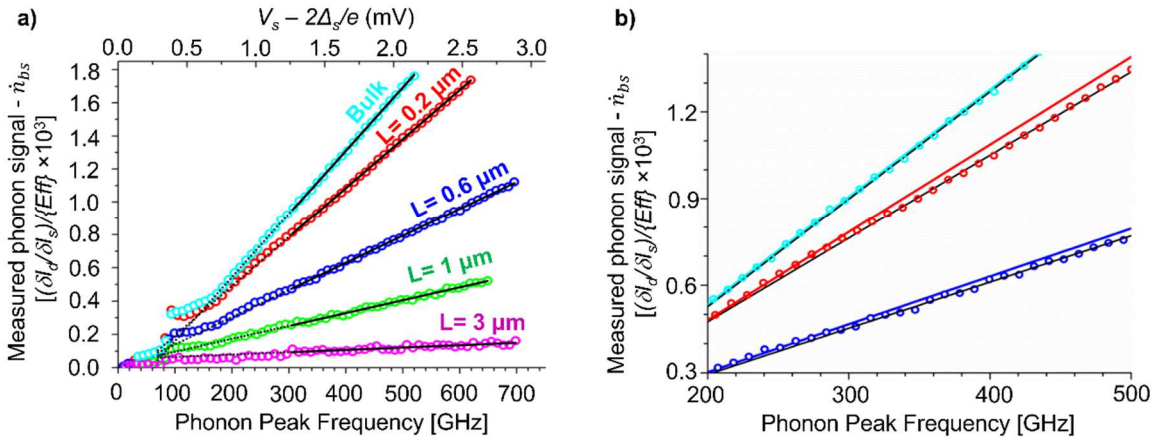
of heights. The roughness  $\eta$  is given by the RMS deviation of  $\sim 1$  nm from the mean amplitude.

A total of ten different silicon-nanosheet arrays were measured (**Table 5.1**). All of these arrays were embedded into  $7\ \mu\text{m}$  wide silicon mesas. In **Figure 5.3a** we present transmission measurements through four arrays of similar cross-section dimensions but differing sheet lengths. Transmission through an un-etched (bulk) Si mesa is also presented for comparison. (In these data, the contribution of substrate back-scatter has been subtracted from the signal, as described below in Eq. 5.4. See **Figure D.1**, Appendix D, for comparison with a spectrometer in which a trench blocks the line-of-sight phonon transmission). These four measured spectra (along with six additional spectra, see **Figure D.2**, Appendix D) exhibit two significant behaviors: 1) Signals show a sharp decrease as the length of the channel is increased. 2) The transmission as a function of phonon peak frequency is very similar to that seen in bulk silicon. Introducing nanosheets into the phonons' path sharply reduces the transmission, but appears to do so independent of frequency. We explore the frequency dependence more closely by trying to quantify the shape of the transmission spectra curves. In **Figure 5.3b**, we apply linear fits in the frequency regimes  $\sim 160\text{-}300$  GHz (colored lines) and  $\geq 300$  GHz (black lines) (See also **Table D.1**, Appendix D). In the bulk Si measurement (**Figure 5.3b**, light blue), both frequency ranges fit well to a single straight line – consistent with the STJ detector's dependence on incident phonon energy (Eq. (3)). Ballistic transmission of phonons through bulk Si is frequency-independent, and various spectrometers measuring through bulk Si present a similar spectral shape (**Figure 5.2d**). A subtly different behavior emerges, however, when measuring transmission through



nanosheets. For phonon peak frequencies above 300 GHz, the signal values clearly lie on a straight line, but below this frequency they fit best to a different line of distinctly higher slope. We can infer from this behavior that phonon transmission through these nanosheets exhibits frequency-dependence only below  $\sim 300$  GHz. At higher frequencies, the strict similarity to the bulk behavior suggests the onset of totally diffusive phonon scattering — the Casimir limit.

The combination of these trends – weak frequency dependence but strong dependence on path length – is consistent with a specularity  $p \ll 1$  and short mean free path ( $l_{eff}$ ) for phonon scattering (Eq. 5.2). The Ziman expression (1), however, does not predict a low specularity throughout this frequency regime: Taking the experimentally-determined roughness  $\eta = 1$  nm, and averaging  $p(\lambda)$  over a phonon spectral distribution with peak frequency at 400 GHz and  $P_{peak}/P_{tot} = 0.32$ ,<sup>43</sup> the mean specularity is  $p = 0.68$ , indicating that the phonons should reflect specularly 68% of the time from this surface.



**Figure 5.3: Phonon Transmission Spectra through Nanosheet Arrays.** a)

Measured transmission signal  $(\delta I_d / \delta I_s) / \{Eff\}$  vs. phonon peak frequency for nanosheet arrays D, E, F, G (see **Table 5.1**) and representative bulk Si sample, after subtraction of measured  $\dot{n}_{bs}$  spectral levels. Comparison to the bulk signal indicates that frequency-dependence of the phonon transmission through nanosheets is small throughout the measured spectral range, and negligible above  $\sim 300$ -400 GHz. The linear fits to each signal are computed for the data above 300 GHz. **b)** Linear fits to the bulk spectrum and two of the nanosheet spectra in a) for regions above and below 300 GHz: The black lines are linear fits for signals  $\geq 300$  GHz while the colored lines are linear fits for signal between  $\sim 160$ -300 GHz.

To more quantitatively predict the signal levels at frequencies above and below the Casimir limit, we must consider the different components of phonon flux arriving at the detector. Below 1 THz, scattering lengths for phonon-phonon and isotope scattering in Si should be much longer than the length of our nanosheets, even exceeding the 500  $\mu\text{m}$  thickness of our sample chip.<sup>23,56</sup> The rate  $\dot{n}_{ph,d}$  of phonons striking the detector therefore comprises four possible components, where  $s$  (source),  $d$  (detector) and  $ns$  (nanosheets) indicate the phonon pathways: (**Figure 5.2e** and Appendix D.1 and D.2.)

$\dot{n}_{los}(s \rightarrow d)$	(line-of-sight transit through bulk Si, frequency-independent)
$\dot{n}_{bs}$	(back-scattering from substrate, possibly freq.-dependent)
$\dot{n}_{los}(s \rightarrow ns \rightarrow d)$	(line-of-sight transit through nanosheets, freq.-independent)
$\dot{n}_{sc}(s \rightarrow ns \rightarrow d)$	(scattering & reflection through nanosheets, freq.-dependent)

Eq. 5.4

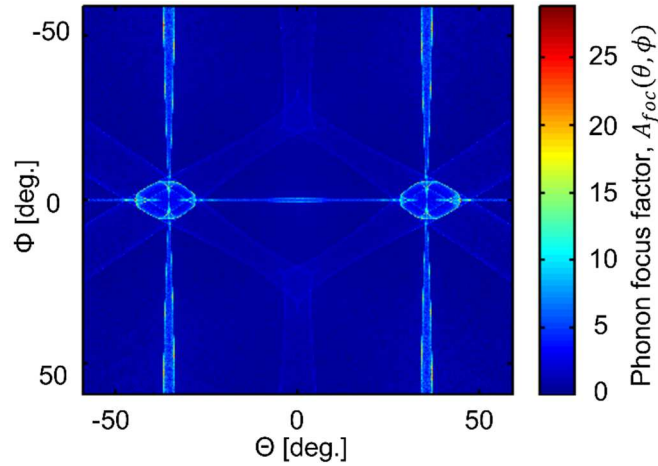
As the quasiparticle diffusion length in aluminum at 0.3 K is of order  $100 \mu\text{m}$ <sup>63</sup>, the measured signal level  $\dot{n}_{bs}$  may include the contribution of phonons that strike the wiring leads far from the junction or the mesa.

In the simplest case, transmission through bulk silicon with no nanosheets, the detected phonon rate should be  $\dot{n}_{ph,d} = \dot{n}_{bs} + \dot{n}_{los}(s \rightarrow d)$ . By defining a line-of-sight ballistic transmission factor  $\Gamma_{los}(s \rightarrow d)$  (having value between 0 and 1), and taking the phonons to be emitted at a single frequency (i.e. assuming for simplicity that  $P_{peak} = P_{tot}$ ), we may express  $\dot{n}_{los}(s \rightarrow d)$  as  $\dot{n}_{los}(s \rightarrow d) = (2\delta I_s/e) \cdot \Gamma_{los}(s \rightarrow d)$ . This bulk transmission factor may be found simply geometrically:

$$\Gamma_{los}(s \rightarrow d) = \int dA_s \int \frac{d\Omega_d}{2\pi} T_{AlSi} \cdot 2 \cdot \cos \psi \cdot T_{SiAl} \cdot A_{foc}(\theta, \phi) \quad \text{Eq. 5.5}$$

Here  $A_s$  is the fraction of the source STJ visible from the detector (i.e., if only 30% of the source STJ lies on the sidewall facing the detector, then we expect only 30% of the phonons produced by  $\delta I_s$  to participate in the measurement). The factor  $2 \cdot \cos \psi$  introduces a ‘Lambert law’ distribution to the intensity of emitted phonons where  $\psi$  is the angle relative to the surface normal of the source STJ sidewall.<sup>63,64</sup>  $T_{AlSi}$  and  $T_{SiAl}$  are acoustic transmission factors of order 1, while integration over  $d\Omega_d$  and  $dA_s$  covers all solid angles subtended by the detector with respect to all visible elements of the source STJ.<sup>49,50,56,65-67</sup> (We note that by properly defining  $dA_s$  and  $d\Omega_d$ , Eq. 5.5 may also be re-stated to express the fraction of phonons transiting from source to detector via specular reflection at the mesa top surface.) The factor  $A_{foc}(\theta, \phi)$  is a ‘phonon focusing’ factor indicating how much the elastic anisotropy in the crystal enhances

( $A_{foc} > 1$ ) or reduces ( $A_{foc} < 1$ ) the phonon intensity along the particular crystal direction  $(\theta, \phi)$ .<sup>66</sup> We find this factor by a MC simulation algorithm – a random  $k$ -vector direction and polarization is assigned to a phonon, and the Christoffel equation is solved for cubic symmetry and the known elastic constants of Si, to find the resulting group velocity (See Appendix D.3, D.4, D.6 and and **Table D.2**, Appendix D).<sup>66,68,69</sup> Repetition over all  $k$ -vector orientations reveals that the phonon flux concentrates along preferred directions. **Figure 5.4** shows a map of phonon focusing factors  $A_{foc}(\theta, \phi)$  centered on the (110) crystallographic direction of Si (See **Figure D.3**, Appendix D, for a similar calculation in the (100) direction, and comparison to literature data). In our measurements,  $A_{foc}$  ranges from 0 to 25 within the integrand of Eq. 5.5.

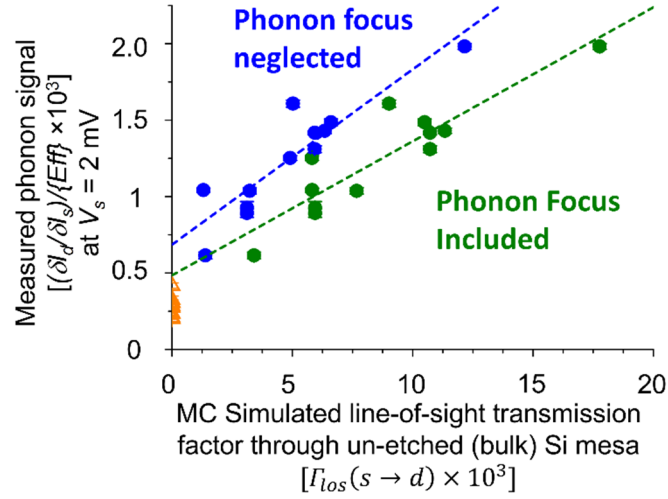


**Figure 5.4: 2D Phonon Focusing Factor Map of Angles Relative to Si [110]**

**Direction Generated by MC Algorithm.** Focusing factors  $A_{foc}(\theta, \phi)$  are presented for angles  $\phi$  and  $\theta$  ranging from  $-59^\circ$  to  $59^\circ$  with average angular resolution of  $\sim 0.12^\circ$ . Angle  $(\theta, \phi) = (0^\circ, 0^\circ)$  is [110] direction, angle  $(\theta, \phi) = (45^\circ, 0^\circ)$  is [100] direction and angle  $(\theta, \phi) = (0^\circ, -90^\circ)$  is [001] direction. In the MC simulations, approximately equal numbers of phonons are assigned to ST, FT and LA

polarizations.

If we experimentally vary  $\Gamma_{los}(s \rightarrow d)$  while keeping all other parameters constant, we expect a linear correspondence between the measured signal and the value of  $\Gamma_{los}(s \rightarrow d)$ . In **Figure 5.5** we present such a measurement. We designed and constructed ten spectrometers transmitting through bulk Si mesas of widths 7, 10 or 15  $\mu\text{m}$  and employing detector fingers of width 3 and 6  $\mu\text{m}$ , so that a large range of transmission factors were represented. All devices were fabricated at the same time on a single silicon wafer, so that all parameters except for  $\Gamma_{los}(s \rightarrow d)$  were held as constant as possible for all ten of the devices. The spectrometer dimensions were verified by inspection in an SEM, to a precision of  $\sim 30$  nm (**Figure 5.1c and 5.1d**). We mounted the chips similarly so that substrate back-scattering levels  $\dot{n}_{bs}$  should also be consistent among all the measurements. We made twelve separate transmission measurements through these devices. Because detector STJ tunnel resistance could not be reproduced precisely from one device to another, we were careful to divide each signal level by the value of  $\{Eff\}$  for that particular detector<sup>51,63</sup> (see also **Table D.3**, Appendix D). According to Eq. 5.3 and Eq. 5.4, the measured detector tunnel currents  $\delta I_d$  should therefore all be proportional to  $\dot{n}_{ph,d} = \dot{n}_{bs} + (2\delta I_s/e) \cdot \Gamma_{los}(s \rightarrow d)$ .



**Figure 5.5: Measured Phonon Transmission Signals through Un-etched (bulk) Si Mesa and Comparison to Simulated Transmission Factors.** Vertical axis is measured signal produced by phonon transmission at  $V_s = 2.0$  mV through un-etched (bulk) Si mesa in ten different geometries. Horizontal axis is MC calculated transmission factor for line-of-sight transmission. Closed circles: measured transmission, comprising backscatter  $\dot{n}_{bs}$  plus line-of-sight  $\dot{n}_{los}(s \rightarrow d)$  contributions. Open orange triangles: independent measurements of backscatter signal  $\dot{n}_{bs}$  on the same sample chips. Vertical error bar at each data point is standard error of average of measured signal value. Green circles: MC simulation includes phonon focusing effects in single crystalline Si. Blue circles: phonon focusing neglected. Dashed blue and green lines correspond to linear fits for blue and green circles.

In **Figure 5.5**, each closed-circle data point represents a measurement of the differential transfer function  $\delta I_d / \delta I_s$  of one device at a voltage of  $V_s = 2.0$  mV (phonon peak frequency  $\sim 390$  GHz). This value is plotted against the calculated transmission

factor  $\Gamma_{los}(s \rightarrow d)$  for that device. For comparison, we also plot the measured signals against an alternate calculation of transmission factors that neglects phonon focusing ( $A_{foc}(\theta, \phi) = 1$  in E1. 5.5). The open-triangle symbols are measurements of  $\delta I_d / \delta I_s$  made at  $V_s = 2.0$  mV, on these same chips, for pairs of source and detector STJs that do not share a line-of-sight. These twelve values, which should offer an independent measurement of  $\dot{n}_{bs}$  at this  $V_s$  level, have an average value  $2.7 \times 10^{-4}$  and standard deviation  $6.1 \times 10^{-5}$ .

The  $\Gamma_{los}(s \rightarrow d)$  transmission factors were calculated according to Eq. 5.5 using a MC algorithm. This approach is based on ray-tracing of phonons within the region enclosed by the device dimensions.<sup>70</sup> The geometric boundaries are set by the mesa dimensions, and location and extent of STJ source and ‘finger’ detector. We assume the top surface and tilted sidewall surfaces of the mesa to be perfectly smooth. The bottom plane of the mesa is assumed to be an ‘open’ surface through which phonons may escape. Each iteration of the algorithm traces the trajectory of a single phonon within the silicon, beginning at the STJ source located on the mesa (111) sidewall. The frequency of the phonon and its corresponding wavevector ( $k$  vector) magnitude in silicon are fixed. Within the area of the STJ source, a point is randomly chosen for the origin of the trajectory. To account for effects of ‘phonon focusing’ on the trajectory, we randomly select the  $k$  vector’s direction from a Lambertian distribution at the source position, and solve the 3-dimensional equations of motion (Christoffel equations) for the group velocity.<sup>56,65,66,69</sup> Since the source plane is not a symmetry plane for the Si crystal, some of the randomly-selected  $k$  vector directions result in phonon group velocities pointing out of the mesa sidewall; in these cases we must reverse the direction

of the calculated group velocity vector. After leaving the source, the phonon's trajectory is traced in a straight line until it reaches a surface. A phonon striking a perfectly smooth surface reflects specularly and continues in a straight-line trajectory. If it strikes the detector sidewall within the boundaries of the detector, the trajectory is terminated and the phonon is added to the tally of detected phonons. If the phonon crosses the bottom plane of the mesa or displaces laterally more than 30  $\mu\text{m}$  away from the source STJ, it is assumed to be out of the system and is counted as an undetected phonon. The MC simulation is repeated 4 million times to find with high accuracy the fraction reaching the detector. (See Appendix D.5, D.7 and **Figure D.4** and **D.5**, Appendix D, for verification that the MC algorithm matches the prediction of a more conventional geometric calculation.)

In **Figure 5.5**, the linear relationship between experimental signal values and the corresponding simulated phonon transmission factors indicates that we have accurately calculated  $\Gamma_{los}(s \rightarrow d)$  in these ten devices, properly accounted for differences in detector tunneling efficiency  $\{Eff\}$ , and held all other relevant parameters constant. A linear-regression fit of measured  $\frac{\delta I_d / \delta I_s}{\{Eff\}}$  against calculated  $\Gamma_{los}(s \rightarrow d)$  finds a slope of  $0.088 \pm 0.013$ , with intercept  $(4.9 \pm 1.2) \times 10^{-4}$  and an R-square goodness of fit of 0.81. The intercept and the independently-measured  $\dot{n}_{bs}$  levels are nearly in agreement, within uncertainties. This result demonstrates that we can model line-of-sight ballistic phonon transport in our system using  $\dot{n}_{ph,d} = \dot{n}_{bs} + (2\delta I_s / e) \cdot \Gamma_{los}(s \rightarrow d)$  as predicted by Eq. 5.4, and implies that  $\dot{n}_{bs}$  may be independently measured and subtracted from phonon spectral data by using non-line-of-sight source/detector pairs



(**Figure D.1**, Appendix D). The small remaining discrepancy between the intercept and the independently-measured  $\dot{n}_{bs}$  may indicate additional phonon scattering processes occurring in the bulk substrate, for instance impurity scattering and/or scattering from crystal defects. We expect such processes to be independent of mesa and nanosheet geometry, so that at worst they would constitute a fixed offset in the estimate of  $\dot{n}_{bs}$ . We may take the uncertainties in the regression fit to represent the maximum systematic error to be expected in determining a phonon transmission factor  $\Gamma$  from a signal value  $\delta I_d/\delta I_s$ . From this error, we may place an upper limit on the spatial resolution achievable with this type of device: For signals traversing a 7 micron wide Si mesa, we can expect to resolve separations as small as 400 nm between adjacent detectors. This resolution could be readily improved by simple changes such as placing the entire source STJ onto the mesa side-wall. **Figure 5.5** also verifies that phonon focus effects are necessary in order to accurately model the line-of-sight ballistic transport. Assuming that phonons propagate isotropically ( $A_{foc} = 1$ ) leads to a poorer match of the intercept to the independently-measured  $\dot{n}_{bs}$  values. Future studies of phonon transmission in finite sized structures should take account of such effects.<sup>71</sup>

The linear behavior appearing in **Figure 5.5** indicates that the contribution of  $\dot{n}_{bs}$  may be readily subtracted from measured signals, and that our models of phonon propagation and of the source and detector STJ behavior predict accurately the relative signal sizes as the transmission factor is varied. In order to predict absolute signal sizes, we must express the differential transfer function ( $\delta I_d/\delta I_s$ ) quantitatively in terms of phonon transmission factor. To achieve this we may first take the differential phonon emission rate in terms of differential emitter current ( $\dot{n}_{ph,s} = 2\delta I_s/e$ ), and multiply it

by the transmission factor  $\Gamma$  (Eq. 5.5) to find  $\dot{n}_{ph,d}$ . We next must use this value of  $\dot{n}_{ph,d}$  to calculate the detector response (Eq. 5.3), integrating across the phonon spectrum (**Figure 5.2c**) to find detector quasiparticle production rate  $\langle \dot{n}_{QP,ph} \rangle$ . Next, to account for the relationship between quasiparticle production rate and differential detector current, we must scale  $\langle \dot{n}_{QP,ph} \rangle$  by the detector tunneling efficiency  $\{Eff\}$ . Finally, we multiply by  $e$  to find  $\delta I_d$ . If we undertake this calculation for a phonon distribution peaked at  $\sim 390$  GHz, we find that  $\frac{\delta I_d}{\delta I_s} / \{Eff\}$  is equal to  $\sim 8$  times the transmission factor. For comparison, the measured slope in **Figure 5.5** is 0.088. Evidently our prediction of absolute phonon flux is missing a scaling factor, which remains constant across all of the devices in this data set because all device parameters were held fixed except for transmission factor  $\Gamma_{los}(s \rightarrow d)$  and detector efficiency  $\{Eff\}$ . We note that nanosheet transmission data analyzed similarly in **Figure 5.6** (as discussed below) has a slope of 0.139 — a remarkably similar number to the one found in **Figure 5.5**, despite the fact that bulk and nanosheet transmission factors were calculated using two models of very different complexity, and that the two sets of measurements were made months apart on different sets of devices resulting from very different microfabrication processes. (The higher slope in **Figure 5.6** may reflect the fact that the detectors used in these measurements employed slightly thicker Al films, leading to higher absorption  $\alpha_{abs}$ .) A likely explanation for the missing scaling factor is that we may be greatly overestimating detector efficiencies  $\{Eff\}$ .<sup>43</sup> Future resolution of this issue would permit accounting for absolute phonon flux.

Having verified for the case of bulk transmission that the experimental signal values  $\delta I_d/\delta I_s$  scale linearly with transmission factor, we can now model the nanosheet transmission factors to shed light on whether the transmission is in the Casimir limit. If nanosheets are present, then we can use Eq. 5.4 to express the expected phonon arrival rate  $\dot{n}_{ph,d}$  at the detector, defining additional transmission factors  $\Gamma_{los}(s \rightarrow ns \rightarrow d)$  and  $\Gamma_{sc}(s \rightarrow ns \rightarrow d)$ .

$$\begin{aligned} \dot{n}_{ph,d}(\omega) &= \dot{n}_{los}(s \rightarrow ns \rightarrow d) + \dot{n}_{sc}(s \rightarrow ns \rightarrow d) + \dot{n}_{bs} + \dot{n}_{los}(s \rightarrow d) \\ &= \left(\frac{2\delta I_s}{e}\right) \Gamma_{los}(s \rightarrow ns \rightarrow d) + \left(\frac{2\delta I_s}{e}\right) \Gamma_{sc}(s \rightarrow ns \rightarrow d) + \dot{n}_{bs} + \left(\frac{2\delta I_s}{e}\right) \Gamma_{los}(s \rightarrow d) \end{aligned}$$

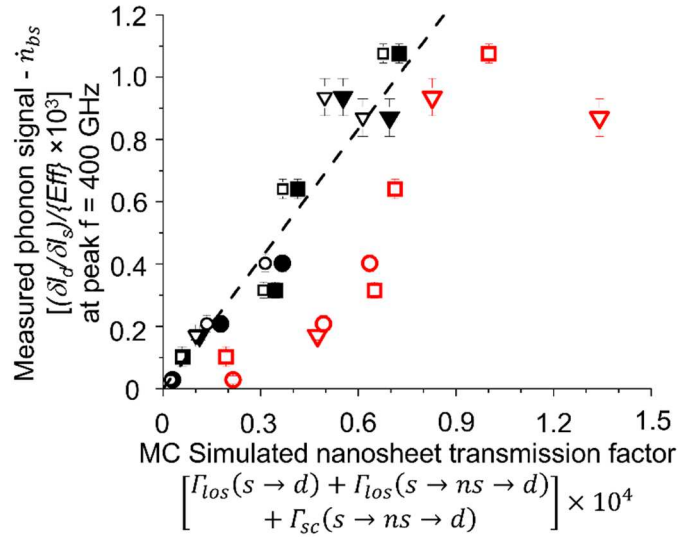
Eq. 5.6

The scattering transmission factor  $\Gamma_{sc}(s \rightarrow ns \rightarrow d)$  will depend on the nanosheet geometry, surface roughness, and source and detector geometries. The last term  $\dot{n}_{los}(s \rightarrow d)$  in Eq. 5.6 accounts for any regions where there may be a path for bulk line-of-sight transport between source and detector, for instance, if the nanosheets are not etched to the full depth of the mesa (0.8  $\mu\text{m}$ ). Then phonons may move ballistically through the un-etched bulk silicon beneath them.

To model the behavior  $\Gamma_{sc}(s \rightarrow ns \rightarrow d)$  of phonons scattering and reflecting multiple times from nanosheet surfaces, we apply our MC ray-tracing algorithm to the more complicated case of a mesa that includes an array of nanosheets. We define dimensions, spacing and position of the array based on SEM images, and include 35 nanosheets in the model. (In our devices, the array size varied from 30 to 40 sheets, but

the 3-5 nanosheets at each end of the array contribute negligibly to the number of detected phonons.) A phonon of frequency  $\omega$  leaving the source is traced in a straight line. Upon striking a nanosheet wall or other surface, its change of motion is modeled according to the specularity ( $p$ ) of the surface. For a given  $\omega$ , we find  $p$  according to the Ziman theory (see Eq. 1), taking 6 km/s as the mean phonon velocity in silicon. AFM measurements of nanosheet sidewalls (**Figure 5.1e**) indicate a roughness of  $\eta = 1$  nm. We assume that the nanosheet sidewall surfaces and the gaps between the nanosheets have this roughness, and that all other surfaces are smooth and have  $p = 1$ . To determine the phonon's change of motion, a random real number ( $s$ ) between 0 and 1 is selected. If  $s < p$ , the phonon reflects specularly at the intersection point. If  $s > p$ , the phonon scatters diffusively: it is re-emitted at the intersection point with a randomized direction that adheres to a Lambertian distribution. (Phonon focusing effects are not taken into account for this re-emission.<sup>23,35</sup>) Upon leaving the surface, the phonon's trajectory is again traced in a straight line until it strikes another surface. This procedure is repeated, with the phonon interacting with as many surfaces as necessary until it is either lost or detected (see **Figure D.6** and **Table D.4**, Appendix D, for test cases for the MC algorithm). For a given phonon frequency, the transmission factor is found as the fraction that strike the detector, out of 4 million trial phonons. Finally, the entire simulation is repeated at a 3 GHz interval across the phonon spectrum (90 to 400 GHz) and these transmission factors are combined in a weighted average based on the phonon power spectral density (**Figure 5.2c**), to find the predicted transmission factor  $\Gamma_{los}(s \rightarrow ns \rightarrow d) + \Gamma_{sc}(s \rightarrow ns \rightarrow d) + \Gamma_{los}(s \rightarrow d)$ .<sup>43</sup>

To compare our measured signal against scattering rates expected in Ziman theory, we measure the differential transfer function  $\delta I_d/\delta I_s$  in ten different nanosheet arrays for a phonon distribution peaked at 400 GHz (**Figure 5.3a** and **Figure D.2**, Appendix D). As in the bulk measurements, we attempted to fashion all devices identically except for their geometries – all spectrometers were fabricated simultaneously onto 7  $\mu\text{m}$  wide Si mesas, and later patterned and etched to form the nanosheets. In this set of measurements, we estimated  $\dot{n}_{bs}$  on each sample chip by independent measurement at the same peak frequency (400 GHz), and subtracted this value from the nanosheet transmission signals. (Estimated uncertainties in this subtraction appear as error bars in our data.) We plot these signal values  $\delta I_d/\delta I_s/\{Eff\}$  in **Figure 5.6** (y-axis) against calculated values of  $\Gamma_{los}(s \rightarrow ns \rightarrow d) + \Gamma_{sc}(s \rightarrow ns \rightarrow d) + \Gamma_{los}(s \rightarrow d)$ .



**Figure 5.6: Measured Phonon Transmission Signals through Si Nanosheet**

**Arrays: Comparison to Modeled Transmission Factors.** Vertical axis is the measured signal for phonon transmission. Horizontal axis is the phonon transmission

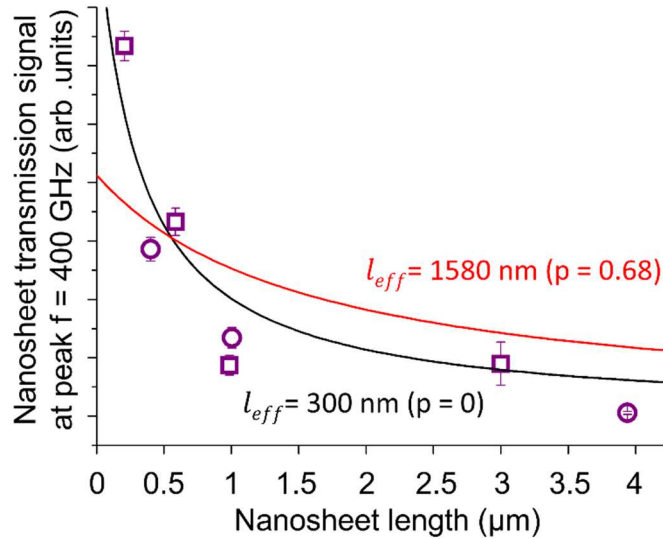
factor calculated using MC algorithm. Circles: arrays A to C (widths  $\sim 0.15 \mu\text{m}$ ). Squares: arrays D to G (widths  $\sim 0.13 \mu\text{m}$ ). Triangles: arrays H to J (widths  $\sim 0.35 \mu\text{m}$ ) (see **Table 5.1**). All measurements employ a phonon distribution peaked at 400 GHz. Vertical error bars derive from standard error of signal average in each measurement, and uncertainties in subtraction of  $\dot{n}_{bs}$ . The model includes phonon focusing effects in single crystalline Si. Phonon transmission factor comprises surface scattering term  $\Gamma_{sc}(s \rightarrow ns \rightarrow d)$ , and line-of-sight transmission terms  $\Gamma_{los}(s \rightarrow ns \rightarrow d)$  and  $\Gamma_{los}(s \rightarrow d)$ , as described in Eq. (6). Red points: simulation employs Ziman expression, Eq. (1), to find nanosheet side-wall specularity, assuming roughness of  $\eta = 1 \text{ nm}$  (as measured using AFM, see **Figure 5.1e**). Black closed points: simulation assumes specularly  $p = 0$  at nanosheet side walls. Black open points: simulation completely neglects nanosheet surface-scattering ( $\Gamma_{sc}(s \rightarrow ns \rightarrow d)$  set to zero). The fraction of phonons transmitted through each nanosheet array is calculated for the estimated phonon frequency distribution (**Figure 5.2c**) of peak frequency 400 GHz, as described in the text. The zero specularly prediction matches the measured signal with high confidence (dashed line: a linear fit to black closed points has  $R^2 = 0.97$  and zero intercept). The prediction of the Ziman model, by contrast, exhibits a poor match to data.

In simulating phonon transmission through nanosheets we try three possibilities. In the first case (**Figure 5.6**, black open symbols), we assume  $\Gamma_{sc}(s \rightarrow ns \rightarrow d) = 0$ , i.e., we neglect any phonons scattering from nanosheet surfaces and consider only phonons that either travel in direct ballistic line-of-sight from source to detector, or

reflect specularly from the mesa top and strike the detector. In the second case (**Figure 5.6**, black closed symbols) we assume specularity  $p = 0$  for the nanosheet sidewalls regardless of the phonon frequency. This condition corresponds to the Casimir-limit regime. In the third case (**Figure 5.6**, red open symbols) we use the Ziman theory of Eq. (1) to calculate the specularity  $p(\lambda)$  for each phonon-surface interaction, as described above. The mesa-top surface comprises un-etched Si, which has a very low roughness (**Figure 5.1e**) and therefore for all three cases this surface is assumed to be purely specular. In the third case, where the phonon transmission should be frequency-dependent, we weight the calculated transmission factors across the estimated STJ emission spectrum. As already shown for the bulk phonon transmission (**Figure 5.5**), a linear relationship between the experimental signal values and the corresponding simulated phonon transmission factors should indicate that the transmission factors are accurately calculated. Additionally, because the background scattering ( $\dot{n}_{bs}$ ) has been subtracted from the experimental signal values, the line should pass through the origin.

Examination of **Figure 5.6** indicates that assuming purely-diffusive sidewalls ( $p = 0$ ) predicts the relative magnitudes of the phonon transmission signal accurately, whereas calculating specularities based on the Ziman expressions and the measured sidewall roughness does not. For instance, consider nanosheets  $\sim 0.12 \mu\text{m}$  wide and  $0.6 \mu\text{m}$  long (Type E, **Table 5.1**). Ziman theory predicts that if the length is increased to  $3 \mu\text{m}$  (type G), the transmitted flux should reduce by  $3.7\times$ , whereas if  $p = 0$  is assumed then theory predicts a reduction of  $6.8\times$ . In fact, in the experiment the signal of type G is reduced by  $6.3\times$ , compared to type E. Or consider nanosheets  $\sim 0.14 \mu\text{m}$  wide and  $1 \mu\text{m}$  long (Type B). Increasing the length to  $4 \mu\text{m}$  (type C) reduces the measured

transmission by  $7.4\times$ , whereas Ziman theory predicts  $2.3\times$ , and the  $p = 0$  assumption predicts  $6.1\times$ . If we take the plot of the zero-specularity prediction vs. measured data, and fit a straight line, we find a slope  $0.139 \pm 0.007$  and a fit confidence (adjusted- $R^2$ ) of 97%, with an intercept consistent with zero, while the Ziman prediction offers a much poorer match and a negative intercept, which is physically incorrect (See **Figure D.7**, Appendix D, for the distribution of specularity values predicted by Ziman theory for the estimated STJ emission spectrum interacting with a 1 nm surface roughness). The zero-specularity predictions, moreover, match the data well for peak frequencies 400, 500 and 600 GHz (**Figure D.8**, Appendix D). This frequency-independent behavior is consistent with ‘Casimir limit’ scattering.



**Figure 5.7: Estimated Transmission Through Si Nanosheet, as a Function of Nanosheet Length, for Phonon Distribution Peaked at 400 GHz.** The signal values shown in **Figure 6** for the nanosheet arrays are corrected here to find a best-estimate of the single-sheet transmission factor, as described in the text. Vertical error bars



derive from those in **Figure 5.6** by the same set of corrections. Circles: arrays A to C. Squares: arrays D to G. (See **Table 5.1**). Also plotted is the predicted transmission fraction  $l_{eff}/(l_{eff} + L)$  for boundary-scattering mean-free paths  $l_{eff} \simeq 0.30 \mu\text{m}$  (black) and  $l_{eff} \simeq 1.58 \mu\text{m}$  (red), corresponding to specularities  $p = 0$  and  $p = 0.68$ , respectively.

While **Figure 5.6** demonstrates that measured signal levels are inconsistent with our model's predictions employing the Ziman theory and the measured surface roughness, it offers little direct insight into the boundary-scattering mean-free path  $l_{eff}$ . Published expressions for the ‘Casimir limit’ mean free path  $l_0$  of a rectangular channel<sup>60,72</sup> differ from our nanosheets in assuming that the channel is isolated and that all four walls have uniform specularity. Nonetheless, using these expressions to provide a rough estimate, the nanosheets in arrays A through G (**Table 5.1**) have Casimir-limit mean free path  $l_0 = 0.26$  to  $0.33 \mu\text{m}$ . For  $p = 0.68$ , we find from Eq. (2) that  $l_{eff} = 1.37$  to  $1.73 \mu\text{m}$ . As a prediction for transmission  $\Gamma$  in our nanosheets, we can use the expression  $\Gamma = l_{eff}/(l_{eff} + L)$ , which models the transport regime where there are both ballistic and diffusive contributions to transmission.<sup>73-75</sup> Whereas this model takes  $\Gamma$  to be the ratio of phonons exiting to those entering a nanosheet, the measured values  $\delta I_d/\delta I_s/\{Eff\}$  in **Figure 5.6** represent a ratio of phonon fluxes measured far from the entrance and exit of an array of nanosheets. Therefore, to compare the data of **Figure 5.6** to the transmission model we must first extract from the data a value  $\Gamma_{sheet}$  for transmission through a single sheet. To account empirically for the demonstrated

inefficiencies in the detection, we divide by the measured line slope  $\sim 0.09$  shown in **Figure 5.5**. To account for un-etched portions of the mesa beneath the array, we subtract an estimate of line-of-sight ballistic transmission  $\Gamma_{los}(s \rightarrow d)$  in each measurement. (See Eq. (6).) Finally, we correct for the line-of-sight ballistic propagation between the STJ source and the nanosheet entrance-faces, and between the nanosheet exit-faces and the detector. Using Eq. (5) and the dimensions listed in **Table 5.1**, a factor is computed for each nanosheet, and these are then summed to find a correction factor for the array. These combined corrections and scaling may be expressed as  $\Gamma_{sheet} = \frac{[\delta I_d / \delta I_s / \{Eff\}] / 0.09 - \Gamma_{los}(s \rightarrow d)}{\sum_i \Gamma_{los}(s \rightarrow ns) \cdot \Gamma_{los}(ns \rightarrow d)}$ , with the sum ranging over all sheets in the array. This correction should accurately extract the relative values of  $\Gamma_{sheet}$  from the measured transmission of different arrays. In principle, a measurement of the  $L \ll l_{eff}$  case (for which the model predicts  $\Gamma = 1$ ) could provide a further absolute scaling. Since this is not available, we adjust the height of model curves as a fit parameter. The result appears in **Figure 7**, along with model predictions of  $\Gamma$  for  $l_{eff} = 0.30 \mu\text{m}$  ( $p = 0$  case) and for  $l_{eff} = 1.58 \mu\text{m}$  ( $p = 0.68$  case). The model curve heights (4.4 and 2.0, respectively) are of order 1 as expected. The match for the  $p = 0$  case is far better, having an adjusted  $R^2$  value of 0.84, whereas the  $p = 0.68$  prediction yields an adjusted  $R^2$  value of 0.44.

In combination with the observations of **Figure 5.3** that show frequency independent spectral structure above  $\sim 300$  GHz, the behavior in **Figure 5.6** and **Figure 5.7** strongly implies that our nanosheets exhibit Casimir-limit behavior at frequencies as low as 400 GHz. A purely line-of-sight prediction ( $\Gamma_{sc}(s \rightarrow ns \rightarrow d) = 0$ ; **Figure 5.6**, black open symbols) offers yet another useful comparison. The data matches this

prediction nearly as well as the diffusive sidewall ( $p = 0$ ) prediction. Evidently, very few phonons that strike the nanosheet side-walls reach the detector, indicating a surface scattering rate far in excess of the Ziman prediction for 1 nm roughness. Geometric effects such as acoustic confinement within the sheets or scattering from impedance-mismatch at the ends of the sheets are unlikely to explain the observed behavior, since our nanosheets are all at least 120 nm wide, significantly larger than the phonon wavelengths.<sup>4,6,76</sup>

If we assume that phonon-surface interactions dominate the phonon trajectory, this result raises the question whether the surface scattering rates could be better predicted by alternatives to the Ziman theory. One notable limitation of the Ziman theory is that the specular constant (Eq. 5.1) is derived only for the case of normal phonon incidence. A more advanced model has been proposed to account for the incident angle of the phonons.<sup>24,77</sup> However, this model predicts that phonons at normal incidence have the lowest specular constant, while at grazing angles, the specular constant will approach one. This more sophisticated model would therefore predict a transmission factor in even greater disagreement with our measurements. Other alternative theories go beyond the Casimir-Ziman model in considering more than just the height of the surface roughness.<sup>2,7</sup> Waves scattering from a surface that exhibits lateral correlations in its roughness structure may exhibit more complicated effects of phase coherence that enhance the degree of scattering. Such theories may offer insight into our nanosheet measurements.

Another question arising from our result is whether additional surface scattering mechanisms may be at play besides the effects of surface roughness. Contamination,

amorphization or excess oxidation of the silicon surface could all reduce its specularly. For instance, recent molecular dynamics simulations predict that adding an amorphous layer to a Si nanowire surface can reduce the wire's thermal conductance more than 70%.<sup>78</sup> In our nanosheet fabrication process, we took precautions to limit damage or contamination of the etched surfaces. During the etch, a polymer coating protected the sidewall surfaces from direct ion bombardment. This coating was subsequently stripped in an oxygen plasma that likewise avoids direct ion bombardment of the sample. Surface oxidation in air remains a possibility, one that would likely apply to many other studies of silicon nanostructures. While our fabrication procedures make it unlikely that phonons are scattering from defects within the bulk silicon, future studies should supplement the AFM roughness measurement with a careful elemental and atomic structural analysis of the nanosheet surfaces and sub-surfaces.

### **5.3. Conclusion**

Our observations of excess phonon scattering rates suggest implications for thermal conductivity in nanostructures.<sup>2,44,79</sup> For instance, for a Si nanowire with  $\sim 1$  nm surface roughness at  $T \sim 5$  K, our result suggests that if surface-scattering dominates all other mechanisms, then the wire's thermal conductance should be 4 times lower than is typically assumed based on Casimir-Ziman theory and the dominant phonon frequency approximation. Even at room temperature, such enhanced surface scattering will strongly impact the thermal conductivity of nanostructures. Recent studies suggest that at 300 K, nearly half of the thermal energy in Si is carried by phonons whose mean

free paths are longer than 1  $\mu\text{m}$ .<sup>39-41</sup> The microscale STJ phonon spectrometer thus offers a new tool to explore important surface-scattering phenomena.

## REFERENCES

- <sup>1</sup> J. B. Hertzberg, M. Aksit, O. O. Otelaja, D. A. Stewart, and R. D. Robinson, Nano Lett. **14** (2), 403 (2013).
- <sup>2</sup> A. I. Hochbaum, R. K. Chen, R. D. Delgado, W. J. Liang, E. C. Garnett, M. Najarian, A. Majumdar, and P. D. Yang, Nature **451** (7175), 163 (2008).
- <sup>3</sup> R. Chen, A. I. Hochbaum, P. Murphy, J. Moore, P. D. Yang, and A. Majumdar, Phys. Rev. Lett. **101** (10), 105501 (2008).
- <sup>4</sup> J. S. Heron, T. Fournier, N. Mingo, and O. Bourgeois, Nano Lett. **9** (5), 1861 (2009).
- <sup>5</sup> J. K. Yu, S. Mitrovic, D. Tham, J. Varghese, and J. R. Heath, Nat. Nanotechnol. **5** (10), 718 (2010).
- <sup>6</sup> J. S. Heron, C. Bera, T. Fournier, N. Mingo, and O. Bourgeois, Phys. Rev. B **82** (15), 155458 (2010).
- <sup>7</sup> K. Hippalgaonkar, B. L. Huang, R. K. Chen, K. Sawyer, P. Ercius, and A. Majumdar, Nano Lett. **10** (11), 4341 (2010).
- <sup>8</sup> J. W. Lim, K. Hippalgaonkar, S. C. Andrews, A. Majumdar, and P. D. Yang, Nano Lett. **12** (5), 2475 (2012).
- <sup>9</sup> J. H. Seol, I. Jo, A. L. Moore, L. Lindsay, Z. H. Aitken, M. T. Pettes, X. S. Li, Z. Yao, R. Huang, D. Broido, N. Mingo, R. S. Ruoff, and L. Shi, Science **328** (5975), 213 (2010).

- 10 D. A. Broido, M. Malorny, G. Birner, N. Mingo, and D. A. Stewart, Appl. Phys. Lett. **91** (23), 231922 (2007).
- 11 A. Ward, D. A. Broido, D. A. Stewart, and G. Deinzer, Phys. Rev. B **80** (12), 125203 (2009).
- 12 K. Esfarjani, G. Chen, and H. T. Stokes, Phys. Rev. B **84** (8), 085204 (2011).
- 13 J. Zou and A. Balandin, J. Appl. Phys. **89** (5), 2932 (2001).
- 14 H. J. Maris and S. Tamura, Phys. Rev. B **85** (5), 054304 (2012).
- 15 N. Yang, X. F. Xu, G. Zhang, and B. W. Li, AIP Adv. **2** (4), 041410 (2012).
- 16 A. L. Moore, S. K. Saha, R. S. Prasher, and S. Li, Appl. Phys. Lett. **93** (8), 083112 (2008).
- 17 A. J. H. McGaughey and A. Jain, Appl. Phys. Lett. **100** (6), 061911 (2012).
- 18 A. A. Maznev, J. Appl. Phys. **113** (11), 113511 (2013).
- 19 G. Chen, *Nanoscale energy transport and conversion : a parallel treatment of electrons, molecules, phonons, and photons*. (Oxford University Press, Oxford; New York, 2005).
- 20 H. B. G. Casimir, Physica **5** (6), 495 (1938).
- 21 K. Fuchs, Math. Proc. Cambridge. **34**, 100 (1938).
- 22 J. M. Ziman, *Electrons and phonons; the theory of transport phenomena in solids*. (Clarendon Press, Oxford, 1960).
- 23 T. Klitsner, J. E. Vancleve, H. E. Fischer, and R. O. Pohl, Phys. Rev. B **38** (11), 7576 (1988).
- 24 S. B. Soffer, J. Appl. Phys. **38** (4), 1710 (1967).
- 25 Z.M. Zhang, *Nano/Microscale Heat Transfer*. (McGraw-Hill, New York,

2007).

- <sup>26</sup> H. J. Lee, J. S. Jin, and B. J. Lee, *J. Appl. Phys.* **112** (6), 063513 (2012).
- <sup>27</sup> J. Cuffe, O. Ristow, E. Chavez, A. Shchepetov, P. O. Chapuis, F. Alzina, M. Hettich, M. Prunnila, J. Ahopelto, T. Dekorsy, and C. M. S. Torres, *Phys. Rev. Lett.* **110** (9), 095503 (2013).
- <sup>28</sup> W. Fon, K. C. Schwab, J. M. Worlock, and M. L. Roukes, *Phys. Rev. B* **66** (4), 045302 (2002).
- <sup>29</sup> Z. J. Wang, J. E. Alaniz, W. Y. Jang, J. E. Garay, and C. Dames, *Nano Lett.* **11** (6), 2206 (2011).
- <sup>30</sup> H. F. C. Hoevers, M. L. Ridder, A. Germeau, M. P. Bruijn, P. A. J. de Korte, and R. J. Wiegerink, *Appl. Phys. Lett.* **86** (25), 251903 (2005).
- <sup>31</sup> V. Yefremenko, G. S. Wang, V. Novosad, A. M. Datesman, J. E. Pearson, R. Divan, C. L. Chang, T. P. Downes, J. McMahon, L. Bleem, A. T. Crites, S. S. Meyer, and J. E. Carlstrom, *IEEE Trans. Appl. Supercond.* **19** (3), 489 (2009).
- <sup>32</sup> N. Mingo, *Phys. Rev. B* **68** (11), 113308 (2003).
- <sup>33</sup> M. Kazan, G. Guisbiers, S. Pereira, M. R. Correia, P. Masri, A. Bruyant, S. Volz, and P. Royer, *J. Appl. Phys.* **107** (8), 083503 (2010).
- <sup>34</sup> D. Lacroix, K. Joulain, D. Terris, and D. Lemonnier, *Appl. Phys. Lett.* **89** (10), 103104 (2006).
- <sup>35</sup> Y. F. Chen, D. Y. Li, J. R. Lukes, and A. Majumdar, *J. Heat Trans.-T. ASME* **127** (10), 1129 (2005).
- <sup>36</sup> L. Shi, *Nanoscale Microscale Thermophys. Eng.* **16** (2), 79 (2012).
- <sup>37</sup> D. G. Cahill, W. K. Ford, K. E. Goodson, G. D. Mahan, A. Majumdar, H. J.

- Maris, R. Merlin, and S. R. Phillpot, *J. Appl. Phys.* **93** (2), 793 (2003).
- 38 T. Klitsner and R. O. Pohl, *Phys. Rev. B* **36** (12), 6551 (1987).
- 39 A. J. Minnich, J. A. Johnson, A. J. Schmidt, K. Esfarjani, M. S. Dresselhaus,  
K. A. Nelson, and G. Chen, *Phys. Rev. Lett.* **107** (9), 095901 (2011).
- 40 K. T. Regner, D. P. Sellan, Z. H. Su, C. H. Amon, A. J. H. McGaughey, and J.  
A. Malen, *Nat. Commun.* **4**, 1640 (2013).
- 41 J. A. Johnson, A. A. Maznev, J. Cuffe, J. K. Eliason, A. J. Minnich, T. Kehoe,  
C. M. S. Torres, G. Chen, and K. A. Nelson, *Phys. Rev. Lett.* **110** (2), 025901  
(2013).
- 42 A. J. Kent, N. M. Stanton, L. J. Challis, and M. Henini, *Appl. Phys. Lett.* **81**  
(18), 3497 (2002).
- 43 O. O. Otelaja, J. B. Hertzberg, M. Aksit, and R. D. Robinson, *New J. Phys.*  
**15**, 43018 (2013).
- 44 A. I. Boukai, Y. Bunimovich, J. Tahir-Kheli, J. K. Yu, W. A. Goddard, and J.  
R. Heath, *Nature* **451** (7175), 168 (2008).
- 45 M. Aksit, D. P. Toledo, and R. D. Robinson, *J. Mater. Chem.* **22** (13), 5936  
(2012).
- 46 J. N. Coleman, M. Lotya, A. O'Neill, S. D. Bergin, P. J. King, U. Khan, K.  
Young, A. Gaucher, S. De, R. J. Smith, I. V. Shvets, S. K. Arora, G. Stanton,  
H. Y. Kim, K. Lee, G. T. Kim, G. S. Duesberg, T. Hallam, J. J. Boland, J. J.  
Wang, J. F. Donegan, J. C. Grunlan, G. Moriarty, A. Shmeliov, R. J. Nicholls,  
J. M. Perkins, E. M. Grievson, K. Theuwissen, D. W. McComb, P. D. Nellist,  
and V. Nicolosi, *Science* **331** (6017), 568 (2011).



- 47 M. Aksit, B. C. Hoselton, H. J. Kim, D. H. Ha, and R. D. Robinson, ACS Appl. Mater. Interfaces **5** (18), 8998 (2013).
- 48 H. Kinder, Phys. Rev. Lett. **28** (24), 1564 (1972).
- 49 J. B. Hertzberg, O. O. Otelaja, N. J. Yoshida, and R. D. Robinson, Rev. Sci. Instrum. **82** (10), 104905 (2011).
- 50 W. Eisenmenger, in *Physical Acoustics - Principles and Methods*, edited by W. P. Mason and R. N. Thurston (Academic Press, New York, 1976), Vol. XII, pp. 79.
- 51 O. Koblinger, J. Mebert, E. Dittrich, S. Dottinger, W. Eisenmenger, P. V. Santos, and L. Ley, Phys. Rev. B **35** (17), 9372 (1987).
- 52 W. Eisenmenger and A. H. Dayem, Phys. Rev. Lett. **18** (4), 125 (1967).
- 53 M. N. Wybourne and J. K. Wigmore, Rep. Prog. Phys. **51** (7), 923 (1988).
- 54 H. Kinder, Z. Phys. **262** (4), 295 (1973).
- 55 M. Welte, K. Lassmann, and W. Eisenmenger, J. Phys. Colloques **33** (C4), 25 (1972).
- 56 S. Tamura, J. A. Shields, and J. P. Wolfe, Phys. Rev. B **44** (7), 3001 (1991).
- 57 S. B. Kaplan, J. Low. Temp. Phys. **37** (3-4), 343 (1979).
- 58 S. B. Kaplan, C. C. Chi, D. N. Langenberg, J. J. Chang, S. Jafarey, and D. J. Scalapino, Phys. Rev. B **14** (11), 4854 (1976).
- 59 Y. S. Ju and K. E. Goodson, Appl. Phys. Lett. **74** (20), 3005 (1999).
- 60 A. K. McCurdy, H. J. Maris, and C. Elbaum, Phys. Rev. B **2** (10), 4077 (1970).
- 61 S. Tamura, Phys. Rev. B **31** (4), 2574 (1985).

- <sup>62</sup> P. E. Hopkins, L. M. Phinney, J. R. Serrano, and T. E. Beechem, Phys. Rev. B **82** (8), 085307 (2010).
- <sup>63</sup> H. J. Trumpp and W. Eisenmenger, Z. Phys. B **28** (3), 159 (1977).
- <sup>64</sup> M. Welte and W. Eisenmenger, Z. Phys. B **41** (4), 301 (1981).
- <sup>65</sup> B. Taylor, H. J. Maris, and C. Elbaum, Phys. Rev. Lett. **23** (8), 416 (1969).
- <sup>66</sup> J. P. Wolfe, *Imaging phonons : acoustic wave propagation in solids*. (Cambridge University Press, Cambridge, U.K.; New York, 1998).
- <sup>67</sup> H. J. Maris, J. Acoust. Soc. Am. **50** (3), 812 (1971).
- <sup>68</sup> K. Jakata, Master's Thesis, University of the Witwatersrand, 2007.
- <sup>69</sup> K. Jakata and A. G. Every, Phys. Rev. B **77** (17), 174301 (2008).
- <sup>70</sup> Christophe Blanc, Ali Rajabpour, Sebastian Volz, Thierry Fournier, and Olivier Bourgeois, Appl. Phys. Lett. **103** (4), 043109 (2013).
- <sup>71</sup> W. Li and N. Mingo, J. Appl. Phys. **114** (5), 054307 (2013).
- <sup>72</sup> Z. Wang and N. Mingo, Appl. Phys. Lett. **99** (10), 101903 (2011).
- <sup>73</sup> S. Datta, *Quantum Transport*, 1<sup>st</sup> ed. (Cambridge University Press, Cambridge, 2005).
- <sup>74</sup> M. J. M. Dejong, Phys. Rev. B **49** (11), 7778 (1994).
- <sup>75</sup> J. A. Wang and J. S. Wang, Appl. Phys. Lett. **88** (11), 111909 (2006).
- <sup>76</sup> K. Schwab, E. A. Henriksen, J. M. Worlock, and M. L. Roukes, Nature **404** (6781), 974 (2000).
- <sup>77</sup> Z. Aksamija and I. Knezevic, Phys. Rev. B **82** (4), 045319 (2010).
- <sup>78</sup> Y. P. He and G. Galli, Phys. Rev. Lett. **108** (21), 215901 (2012).
- <sup>79</sup> J. Carrete, L. J. Gallego, L. M. Varela, and N. Mingo, Phys. Rev. B **84** (7),

075403 (2011).

## CHAPTER 6

### 6. CONCLUSIONS AND OUTLOOK

This work was devoted to synthesis, characterization and simulation of nanosheets and nanostructured thin films. Novel methods were described for scalable and cost-effective fabrication of metal oxide nanosheets and nanostructured thin films. Millimeter long metal oxide nanosheets were self-assembled into stacks with a unique turbostratic arrangement of multi-atomic layer crystal sheets. The metal oxide nanosheets and nanostructured thin films were also investigated for their practically important properties such as mechanical flexibility, electrical conductivity and/or optical transparency. Thermal transport in Si nanosheets were measured using a novel micro-scale phonon spectrometry technique and results from the measurements were compared with phonon transport MC simulations which, unlike most of the previous studies, included crystal anisotropy effects.

#### ***6.0. Millimeter Length Metal Oxide Nanosheets Synthesis***

The metal oxide nanosheet synthesis method is a sol-gel-based, bottom-up process that produces tens of thousands of nanosheet layers packed into a macro-scale pellet. So far,  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets have been fabricated with this low-cost, environmentally benign, and scalable technique. The synthesis method relies on a solution chemistry that should be applicable to other layered oxides. The nanosheets produced with this method are uniform in length and shape with highly anisotropic dimensions of nanometer thickness and millimeter lateral lengths. The majority of previously proposed techniques rely on chemical exfoliation of bulk

materials to produce nanosheets, but this aggressive chemical process can damage or alter the starting compound.<sup>1</sup> For example, chemical exfoliation of thermoelectric bulk  $\text{Na}_x\text{CoO}_2$  results in  $\text{CoO}$  nanosheets,<sup>2</sup> which are insulating and not useful as thermoelectric materials. In our work,  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets are readily delaminated into free-standing, electron-transparent nanosheets with lateral lengths of up to 350 microns, while maintaining their atomic structure. This 350 micron sheet-length marks an order of magnitude improvement over the typical exfoliation methods for metal oxide nanosheets.

The metal oxide nanosheets study involves multiple major novelties of scientific benefit. The low-cost and scalable synthesis of millimeter-length nanosheets is promising for nanoscale thermoelectrics where thermoelectric efficiency is enhanced due to phonon confinement and scattering.  $\text{Na}_x\text{CoO}_2$ , in bulk form, has been vigorously researched recently, and access to nanometer-thick structures of this material should open up new avenues of physics investigations for nano-size induced properties. Both  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets are ductile and conductive. This makes them promising candidates for flexible oxide electronic applications and electro-mechanical sensors. Regarding the synthesis procedure, this is the first time that the Electric-field induced kinetic demixing has been observed to result in new phase formation and to cause an abrupt transition in the atomic concentration. The demixing process should prove to be a novel method for low cost single crystal growth. Moreover, the anisotropic grain growth by the enhanced alkali content is a new pathway for the field of nanomanufacturing.

In this study, the  $\text{Na}_x\text{CoO}_2$  nanosheets did not indicate phonon confinement induced enhancement in thermoelectric properties according to the preliminary thermoelectric measurements. The preliminary measurements indicate that thermal conductivity of the single  $\text{Na}_x\text{CoO}_2$  nanosheet stack is slightly higher than thermal conductivity of single crystalline  $\text{Na}_x\text{CoO}_2$  and its electrical conductivity is order of magnitude lower compared to the electrical conductivity of single crystalline  $\text{Na}_x\text{CoO}_2$ . The reasons for this behavior is not clear and can be associated with experimental mistakes in the sample preparation and preliminary thermoelectric measurements. The nanosheet stacks can be damaged during their mechanical extraction from the macro-scale calcined pellets and they are only  $\sim 1.8$  mm long and  $\sim 100$   $\mu\text{m}$  thick causing difficulties in preparing reliable thermoelectric measurement setups. Understanding the thermoelectric behavior of the  $\text{Na}_x\text{CoO}_2$  nanosheets requires further investigations based on reliable thermoelectric measurements and detailed characterization of the inner microstructure of the nanosheet stacks that are used in thermoelectric measurements.

Future research initiatives for the metal oxide nanosheets study can be listed as follows:

- Because of the perfect registry of the stacked nanosheets along their c-axes, phonon-surface scattering and phonon-confinement effects at the nanosheet boundaries may not be strong enough for significant reduction in the thermal conductivity. Disturbing this perfect ordering with mechanical or chemical

treatment can provide more reduction in thermal conductivity and lead to higher thermoelectric efficiency.

- The mechanism of low temperature e-field induced kinetic demixing is still largely unknown and increasing the understanding of this phenomenon can improve the quality and anisotropy of nanosheet crystals: 1) Time dependent atomic concentrations in different regions of the e-field applied pellet can be investigated with in-situ and ex-situ XRD and EDS characterization. Such analysis can explain how the new phases form and how the abrupt transition in the atomic concentration occurs during the EIKD process. 2) The movement of oxygen sublattice can be observed using Pt nanoparticle tracers. 3) EIKD experimental temperature can be varied between extreme cold ( $\sim 70$  K) and extreme hot ( $\sim 1200$  K) and the effect of temperature on the kinetic demixing rate and homogeneity can be observed. 4) Oxygen partial pressure in EIKD experiments can be carefully tuned to completely understand the role of oxygen gas in EIKD. 5) Instead of oxygen, reactive gas phases of other materials (e.g.,  $F_2$ ,  $Cl_2$ ,  $H_2$  and  $N_2$ ) can be introduced to the EIKD environment to see if these gasses will be involved in EIKD and if novel phases can be produced with this experiment. 6) Metallic elements other than K, Na, and Co can be tried in EIKD as demixing species. 7) Thinner pellets can be used in EIKD experiments to observe if it will improve the homogeneity of the demixing process as oxygen will be delivered to the inner regions of the pellet more quickly.
- Reducing the thicknesses of pellets in EIKD experiments to typical single

nanosheet stack thickness ( $\sim 100\text{ }\mu\text{m}$ ) may also lead to formation of only a single nanosheet stack after the calcination step rather than formation of a macro-scale pellet with many nanosheet stacks in it. Such single nanosheet stack samples would not require the mechanical extraction step which can cause unwanted deformation in the nanosheet stacks.

- The metal oxide nanosheet synthesis method in general can be used to try producing metal oxide nanosheets other than  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$ . These novel nanosheets can be metal oxide nanosheets containing both Na, K and Co and/or metal oxide nanosheet of other practically interesting layered compounds such as  $\text{Na}_{0.5}\text{K}_{0.5}\text{NbO}_3$ ,  $\text{KNbO}_3$ ,  $\text{K}_4\text{Nb}_6\text{O}_{17}$  and  $\text{K}_4\text{Nb}_6\text{O}_{17} \cdot 3\text{H}_2\text{O}$ .
- The porous cobalt oxide region that form during EIKD can be further investigated for practical applications. The porous region is typically a homogenous network of nano-scale metal oxide particles and can be useful as battery electrodes or catalytic applications. Porous nano-crystal networks of other metal oxides such as nickel and manganese oxides can also be obtained using EIKD.
- According to the SEM images, the conductive metal oxide nanosheets can be easily bent under mechanical force. However, the effect of the mechanical bending on the electrical conductivity is still unknown. If the electrical conductivity strongly depends on the mechanical bending, the nanosheets can be utilized as nanoscale mechanical sensors. If the electrical conductivity can be maintained high regardless of the mechanical deformation, the nanosheets can be readily used as oxide conductors in flexible nano-electronics.



- As expected from most of the ceramics,  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  are not known as flexible and ductile materials in bulk form. The reason for ductility and/or flexibility in the nanosheet form can be investigated in detail. The mechanical properties of nanosheets such as yield strength and fracture strength can be measured and compared with bulk values. The mechanical properties for different nanosheet thicknesses can be measured to see if the nanosheet stacks are exhibiting so called super-plastic behavior which is often observed in polycrystalline materials with ultrafine grain size.
- High temperature synthesized metal oxides are usually chemically stable materials. The chemical stability of  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheets can be tested in highly acidic and basic environments. If these materials are chemically resistant, they can be ideal candidates for electrically conductive supports in electro-catalytic applications such as fuel cells.

#### **6.1. *P-type TCO Thin Films of Misfit layered $\text{Ca}_3\text{Co}_4\text{O}_9$***

The sol-gel process from the metal oxide nanosheet synthesis technique is combined with spin coating and low temperature calcination to produce p-type TCO thin films of misfit layered  $\text{Ca}_3\text{Co}_4\text{O}_9$ . The highest visible range optical transmission and lowest sheet resistance for the TCO films are 67.1% and 5.7 k $\Omega$ /sq, respectively. The near infrared region optical transmission is as high as 85% even for the most conductive  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin film. Because the high optical transmittance and electrical conductivity are conflicting properties, the TCO films were evaluated based on their FOM. The top-performing TCO thin film performs significantly better than all other solution processed p-type TCO thin films and better too than most of those

prepared by PVD and CVD. Exceeding the performance of PVD and CVD prepared p-type TCO thin films can be considered as a milestone for commercializing p-type TCOs because the solution-spin coat-based technique is cheaper and more scalable than these other two vacuum methods.

High FOM p-type TCO thin films will be important for energy devices and optoelectronics. The p-type TCO thin films can serve as critical components for various technological developments such as efficient charge injection layers for solar cells with better band matching current collectors, OLEDs and invisible circuits with oxide p-n junctions that require p-type counterparts to the n-type TCOs (e.g., ITO). Very high infrared region transparency of the p-type  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin films is very desirable for applications in near infrared optoelectronics where n-type TCOs provide poor optical transmission.

Invention of p-type  $\text{Ca}_3\text{Co}_4\text{O}_9$  transparent conductor introduces misfit layered oxides in p-type TCO research, which is mostly dominated by delafossites such as  $\text{CuAlO}_2$ .<sup>3</sup> This work is the first to report transparent conductivity in misfit layered  $\text{Ca}_3\text{Co}_4\text{O}_9$  which has been previously known as a remarkable p-type thermoelectric material. It is likely that this work will inspire other researchers in the field to produce high performance p-type TCOs from misfit layered oxides.

In the p-type  $\text{Ca}_3\text{Co}_4\text{O}_9$  TCO thin film study, the electrical conductance of the films are evaluated based on their sheet resistance and the p-type behavior is proved

by Seebeck measurements. Additional resistivity and carrier type measurements can significantly improve this study. These measurements can be achieved by Van der Pauw technique.

The p-type TCO thin film study can be improved by research strategies listed below:

- Nanostructured thin films of other p-type metal oxides can be produced with the same synthesis method. With their higher bulk electrical conductivity  $\text{Ca}_3\text{Co}_{4-x}\text{Cu}_x\text{O}_9$ ,  $\text{Na}_x\text{CoO}_2$  and  $\text{K}_x\text{CoO}_2$  are promising candidates for such studies.<sup>4-6</sup>  $\text{Ca}_3\text{Co}_4\text{O}_9$  can also be doped with K and Na ions to find more conductive alternatives to  $\text{Ca}_3\text{Co}_4\text{O}_9$ .
- Multiple layers of calcined TCO thin films can be deposited on top of each other to improve conductivity. This process may involve thin film layers from different metal oxide compounds. In order to increase the durability of the TCO thin film multilayer structure, the most air-stable compound can be deposited as the top layer.
- Diffusion of atomic species to the quartz substrate can cause difficulties in optimizing the p-type TCO thin film performance because stoichiometry of the final compound can sometimes be significantly different than the relative concentrations of the starting solution. One can overcome such a problem by heating the spun film only by passing AC or DC current through it. The magnitude of the current should be carefully tuned to minimize atomic diffusion to the substrate but maximize crystallization and sintering rate.

- The contact resistances between the nano-crystals in the TCO thin films are likely the major component in the measured resistance of the films. This resistance can be reduced by sintering of the nano-crystals by heat treatment. Such sintering can be most efficiently performed using AC or DC current because applied current will generate most of the heat at the high resistance contacts between the nano-crystals.
- In situ measurement of sheet resistance during the calcination process can help optimize the calcination time and temperature to obtain the highest film conductivity.

## **6.2. *Micro-scale Phonon Spectrometry through Si Nanosheets***

Micro-scale phonon spectrometry technique aims to investigate phonon surface scattering in nanostructures. Determining phonon-scattering rates is important to understanding thermal conductivity in nanomaterials which can contribute to future enhancements in thermoelectric conversion, heat pipes, and thermal insulation. Anomalous low thermal conductances are previously reported for semiconducting and insulating nanowires, and such reduction in thermal conductance is attributed to scattering of phonons from the nanowire surfaces.<sup>7-9</sup> However, the exact mechanism for phonon-surface scattering is not understood in previous studies due to the limitations in experimental diagnostics. In this work phonon transport in Si nanosheets is measured using a microscale phonon spectrometer based on STJs. This technique allows direct measurement of the surface scattering rate of phonons. The results from nanosheet phonon spectrometry experiments diverge from the well-known classical theories.

Previous micro and nanoscale investigations of phonon dynamics have relied on measurements of total thermal conductance.<sup>7-9</sup> As Planck distribution of phonon energies has a large breadth in frequency space, it does not provide a precise probe of phonon scattering rates. The novel micro-scale STJ-based phonon spectrometer, is capable of measuring a much narrower bandwidth of phonon frequency. The high frequency resolution technique can directly correlate nanoscale phonon transmission to phonon frequency in ways previous work couldn't. The micro-scale phonon spectrometer is at least ten times more precise than thermal conductance measurements.

Phonon transmission rates through 120 to 380 nanometers wide silicon nanosheets are measured using the micro-scale phonon spectrometer. The experiments are performed at temperatures below 1 K where phonon-phonon and phonon-impurity scattering may be neglected. This way the phonon spectrometer emits and receives the phonons in ballistic trajectories so that scattering within the nanosheets can be isolated and studied in detail. In order to be able correlate phonon surface scattering to nanosheet surface characteristics, careful measurements of the nanosheet surface roughness are performed using atomic-force microscopy. MC simulations are performed to model ballistic phonon transport and phonon-surface interactions in our experiments. Unlike majority of the previous ballistic thermal transport simulations, crystal anisotropy effects are also included in the MC simulations reported in this study. Comparison of the spectrometer measurements to the MC simulations show that

the measured rates of phonon transmission through the Si nanosheets are lower than what is predicted by well-known Ziman theory for the measured surface roughness. For example, according to Ziman theory, 3 micron long by 120 nm wide nanosheets should have  $\sim 3.7\times$  less phonon transmission than 0.6 micron long nanosheets with the same width. However, measured transmission signals for these nanosheets differ by a factor of 6.3. If instead a maximum phonon-surface scattering probability – the so-called ‘Casimir Limit’ – is assumed in the MC simulations, the relative magnitudes of the signals can be accurately predicted.

Because the work presented in this dissertation mainly focuses on understanding the phonon spectrometry experimental results using the analytical models and MC simulations, future routes are listed only for the modeling and simulation:

- Currently the MC simulations does not provide information regarding time of flight for the transmitted phonons. The MC algorithm is already capable of determining the phonon group velocities for different polarizations along different crystal directions. Incorporating these outputs to phonon transport simulations can provide phonon time of flight information which can be useful for calculating the low temperature thermal conductivity of the nanostructures.
- The MC Simulations can provide low temperature thermal conductance through nanostructures if the frequency distribution of emitted phonons in the simulations is designed to be equivalent to Planck distribution.
- The Monte Carlo simulations can be designed to predict phonon transport at high temperatures by assuming a temperature dependent phonon-phonon

scattering mean free path. If the phonon propagates distances equal to this mean free path the phonon direction can be randomized to imitate the phonon-phonon scattering.

- Phonon-surface interactions in the Monte Carlo simulations is simplified as being either reflective or randomly diffusive. The MC simulations can be significantly improved by incorporating advanced phonon-surface interaction models.
- The MC simulations are only designed for rectangular prism shaped nanostructures. The MC algorithm can be improved to model other 3D shapes, and then use this to determine novel phonon interactions.
- Phonon focusing preferentially depletes phonons along certain crystallographic directions and concentrates them along other directions. Number of phonon-surface interactions in a single crystalline transport medium can be increased if the transport medium is shaped such that phonons are focused along the directions in which the transport medium has the smallest size. Such single crystalline transport medium designs can be readily simulated using the current MC code. For instance a nanowire can be designed to be parallel with [001] or [011] direction and the phonon transport difference between the two cases can be compared.
- Phonon confinement effects can be incorporated in MC simulations by using nanoscale phonon-dispersion relations to calculate the phonon group velocities. If the size of the Si transport medium is smaller than 30-40 nm, one can not assume the relation between the phonon frequency and wave-vector to

be linear and phonon group velocities should deviate from bulk values. These nanoscale phonon group velocities should be used in MC simulations to be able to accurately predict the nanoscale thermal conductivity. Additionally, phonon wavelengths should be determined based on the nano-scale phonon dispersion relations for more accurate calculations of phonon-surface scattering rates.

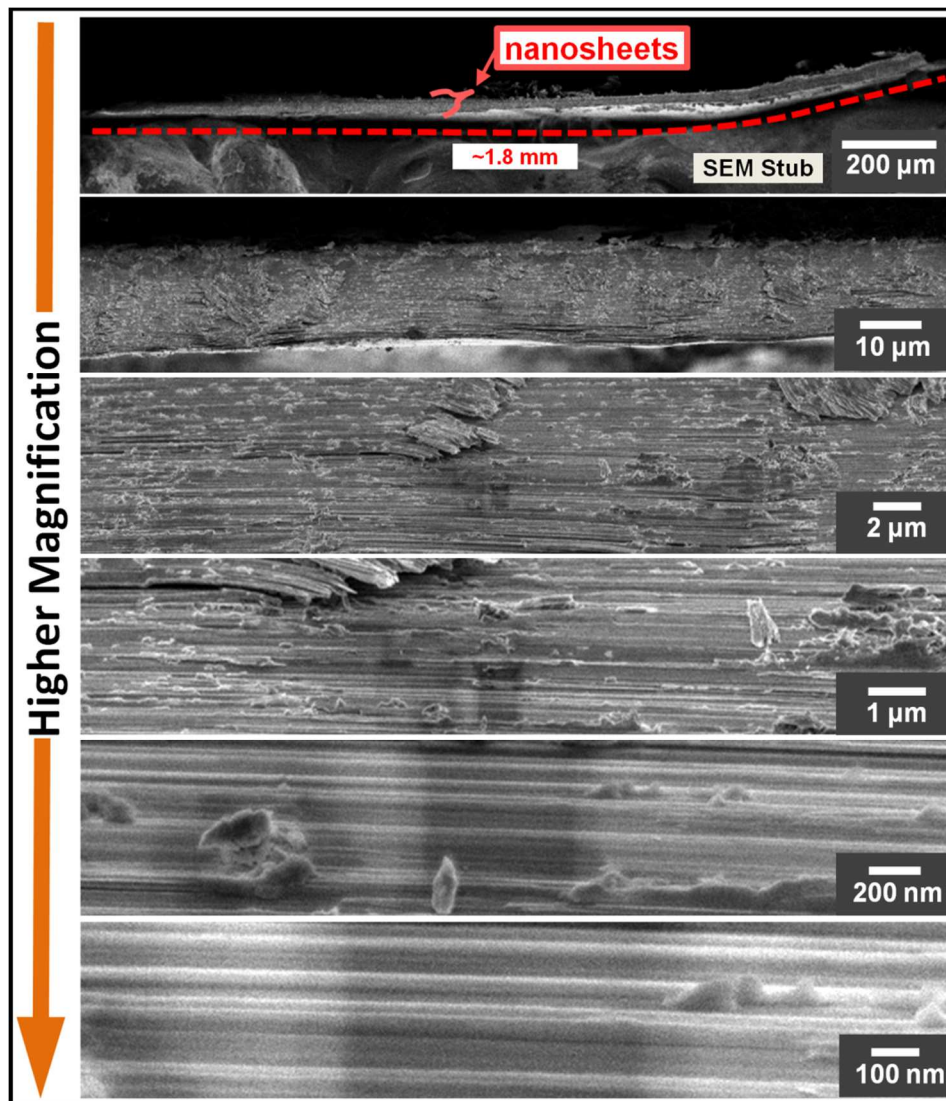


## REFERENCES

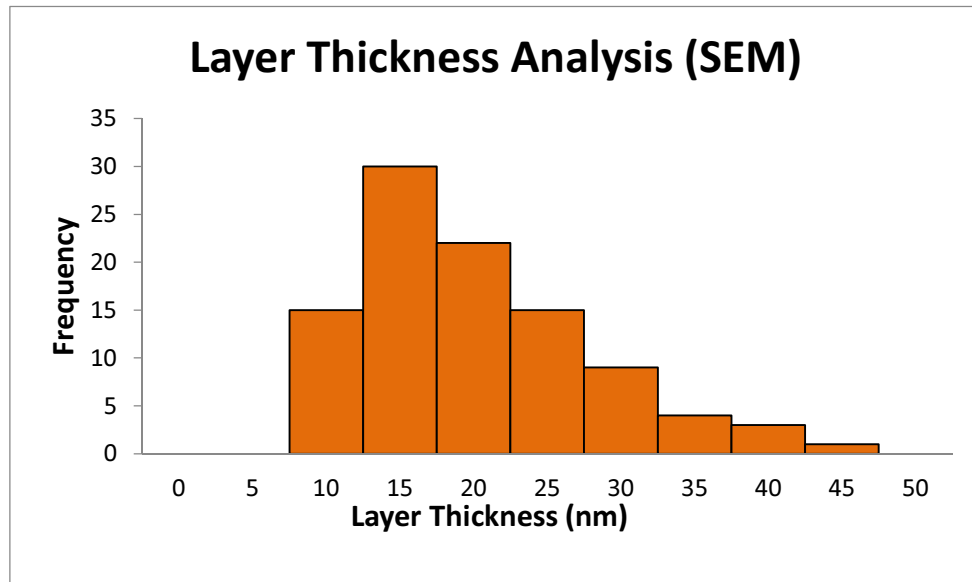
- <sup>1</sup> M. Osada and T. Sasaki, *J. Mater. Chem.* **19** (17), 2503 (2009).
- <sup>2</sup> Y. Masuda, Y. Hamada, W. S. Seo, and K. Koumoto, *J. Nanosci. Nanotechnol.* **6** (6), 1632 (2006).
- <sup>3</sup> J. Tate, M. K. Jayaraj, A. D. Draeseke, T. Ulbrich, A. W. Sleight, K. A. Vanaja, R. Nagarajan, J. F. Wager, and R. L. Hoffman, *Thin Solid Films* **411** (1), 119 (2002).
- <sup>4</sup> Y. A. Huang, B. C. Zhao, X. B. Hu, S. Lin, R. Ang, W. H. Song, and Y. P. Sun, *Dalton Transactions* **41** (36), 11176 (2012).
- <sup>5</sup> I. Terasaki, Y. Sasago, and K. Uchinokura, *Phys. Rev. B* **56** (20), 12685 (1997).
- <sup>6</sup> H. Y. Tang, H. Y. Lin, M. J. Wang, M. Y. Liao, J. L. Liu, F. C. Hsu, and M. K. Wu, *Chem Mater* **17** (8), 2162 (2005).
- <sup>7</sup> K. Hippalgaonkar, B. L. Huang, R. K. Chen, K. Sawyer, P. Ercius, and A. Majumdar, *Nano Lett.* **10** (11), 4341 (2010).
- <sup>8</sup> K. Schwab, E. A. Henriksen, J. M. Worlock, and M. L. Roukes, *Nature* **404** (6781), 974 (2000).
- <sup>9</sup> J. S. Heron, T. Fournier, N. Mingo, and O. Bourgeois, *Nano Lett.* **9** (5), 1861 (2009).

## APPENDIX A

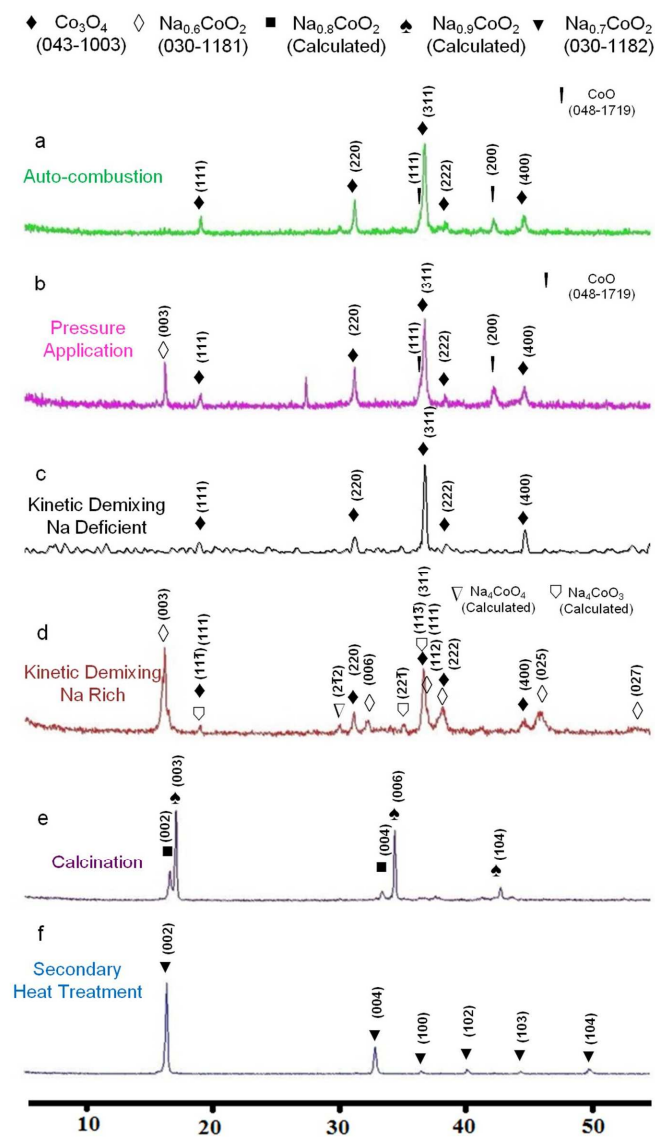
### A. SUPPLEMENTARY INFORMATION FOR CHAPTER 2



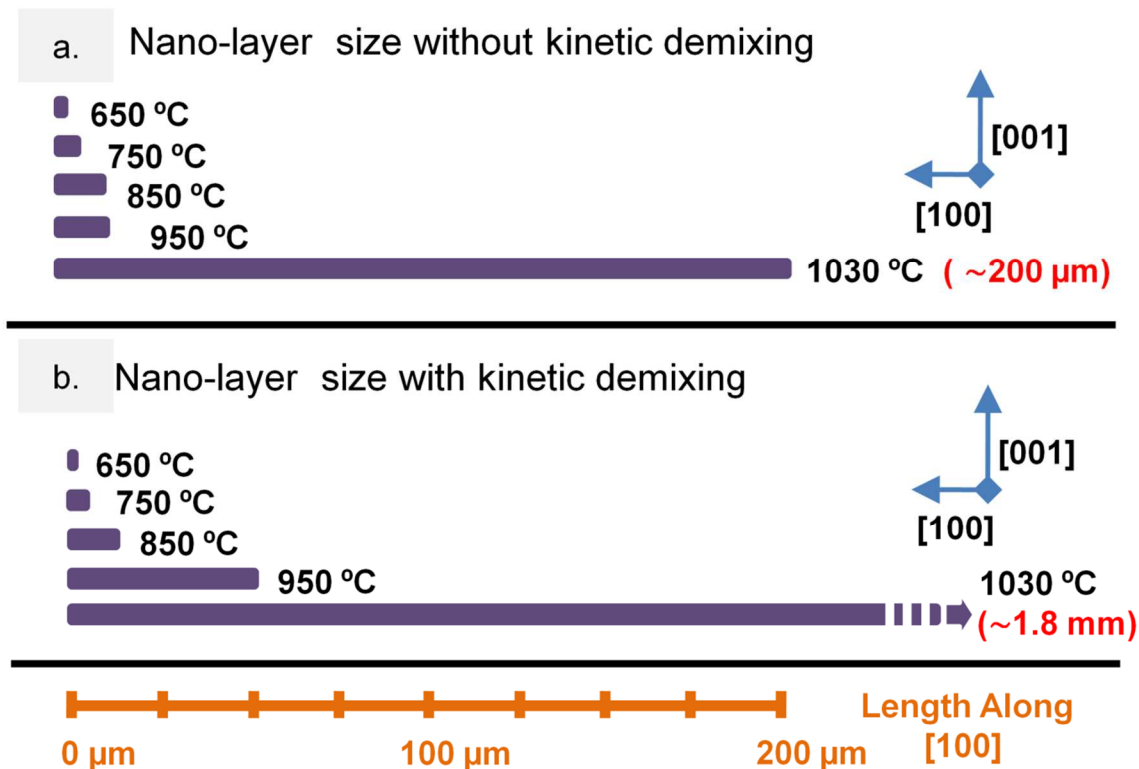
**Figure A.1: Additional SEM images for nanosheet metal-oxides.** All images show the cross-sections of the stacked nanosheet. Images are sorted from lower magnification to higher magnification.



**Figure A.2: Histogram of thickness of 100 layers as measured by SEM.** The average thickness was found to be 18.2 nm with a median thickness of 16 nm and standard deviation of 7.9 nm.



**Figure A.3: X-ray powder diffraction of the  $\text{Na}_x\text{CoO}_2$  between each step of the synthesis procedure.** The samples in the form of pellets and single crystals were ground before the measurement. The peaks are identified for different phases and the available PDF numbers are listed for the phases.

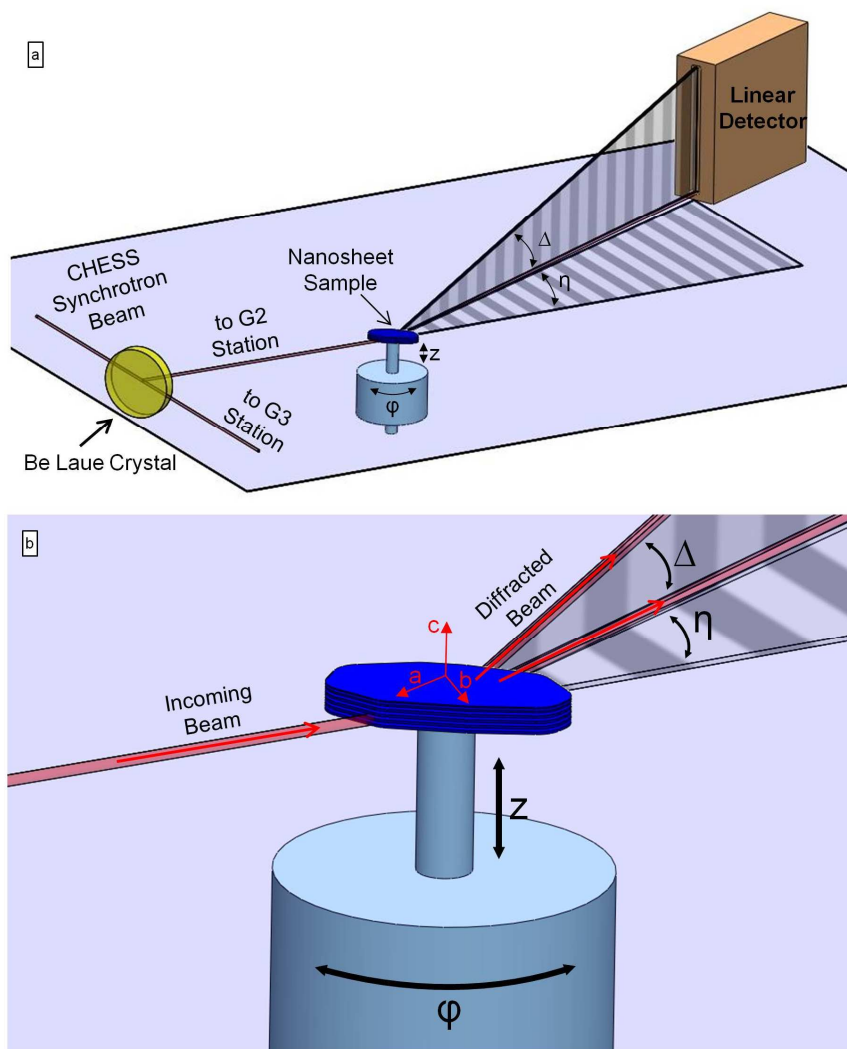


**Figure A.4: Schematic diagram representing anisotropic growth of nano-layers at different calcination temperatures without kinetic demixing (a) and with kinetic demixing (b).** All samples were initially heated up to 650 °C and held for 4 hours. The temperature was then rapidly increased up to different values indicated in the figure and calcined for 3 hours. Under both conditions with and without the kinetic demixing significant growth along c axis [001] is not observed. The layers are all at the same thickness around 20 nm. The layers are much longer with kinetically demixed samples. (The bar at the bottom of the figure is to scale with the sizes above)

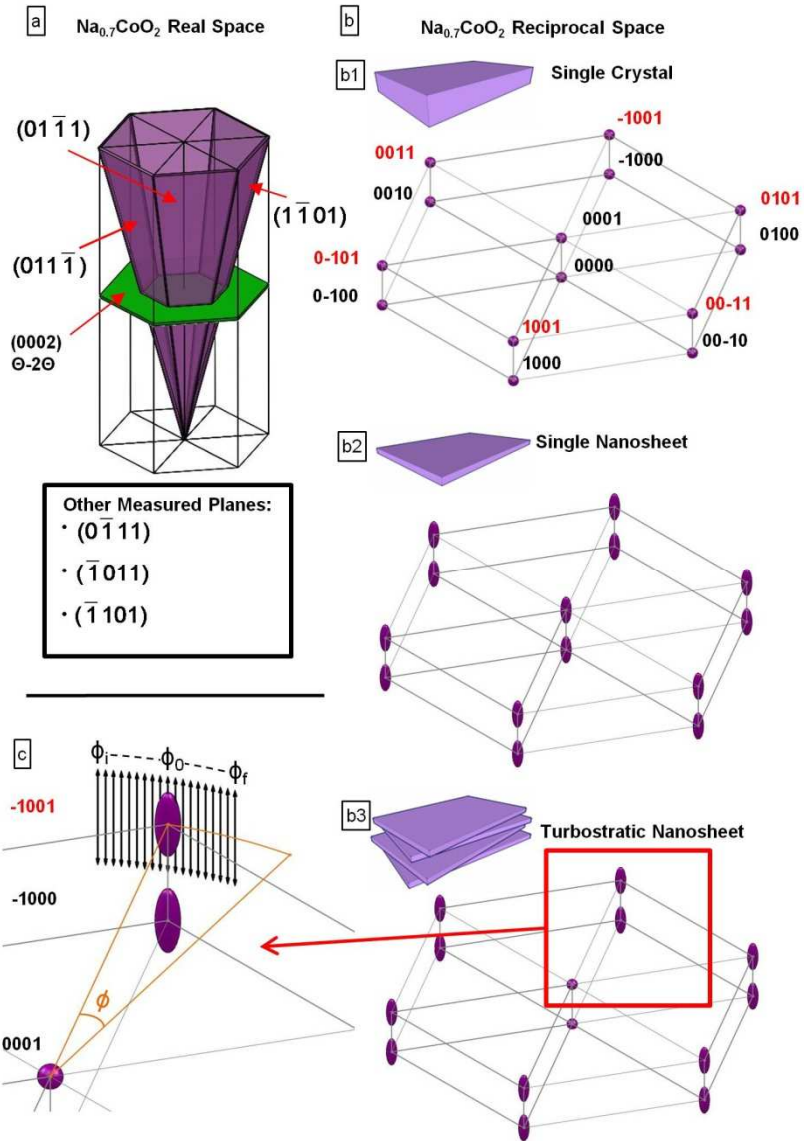
## **CHESS G2 Hutch Background**

A 0.1% bandwidth slice of the intense beam from the CHESS G-line 50-pole wiggler and multilayer monochromator is deflected into the G2 station by a Be single-crystal beam splitter. The incident beam is collimated vertically by certain number of slits as 1 mm, while the full horizontal beam width of about 2 mm is accepted. The horizontal diffractometer has a motorized sample height stage to precisely align the sample surface into the beam. The scattered beam is detected by a linear gas detector after passing a Soller collimator. Collimator and detector have matching apertures of 8 mm horizontal by 100 mm vertical.

For the measurements performed in CHESS G2 Hutch, samples smaller than the beam size are used to increase the penetration depth. Due to the small size of the sample and also the high level of surface roughness, it is expected that the incident beam enters into the material with angles close to  $90^\circ$  at the edges (parallel to (001) planes). It is also known that the beam leaves the material at  $\sim 7.5^\circ$  in order to satisfy the diffraction condition for (10 $\bar{1}$ 1) plane of  $\text{Na}_{0.7}\text{CoO}_2$ . The attenuation length was calculated to be  $\sim 2.5 \mu\text{m}$  at  $7.5^\circ$  for the utilized beam energy (8.65 keV) which is equal to the penetration depth relevant to the number of sampled layers.



**Figure A.5: (a) Schematic representation of the Grazing Incidence Diffraction setup in G line at CHESS.** The incident beam comes in at nearly  $0^\circ$  and the diffracted beam is detected by a linear detector which covers  $10^\circ$  along  $\Delta$  direction.  $\eta$  is set to the  $2\theta$  position on the horizontal crystallographic plane. The sample rotates around the  $\phi$  axis during the scan. (b) A closer look at the sample-beam orientation indicating nanosheets lying parallel to the incident beam.

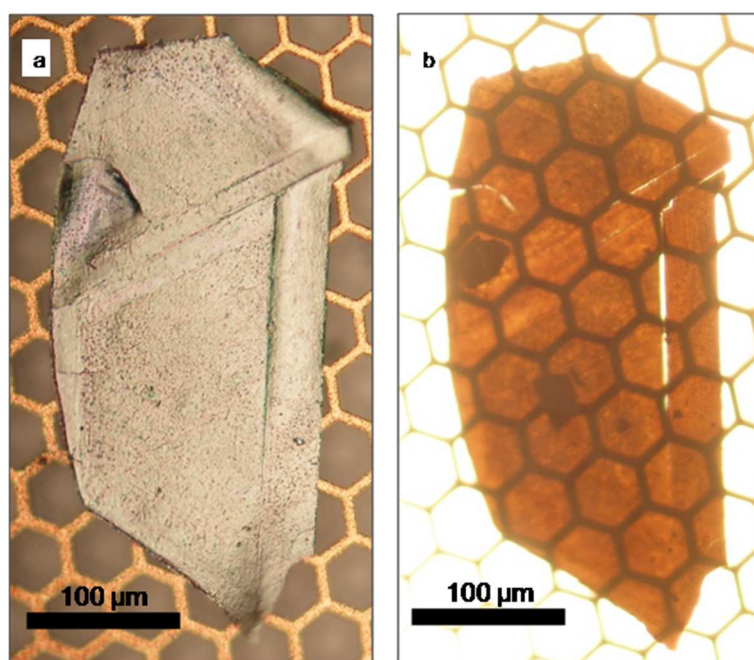


**Figure A.6: Schematic of the real (a) and reciprocal (b) space representation of the  $\text{Na}_x\text{CoO}_2$  lattice.** The red arrows in (a) point to  $(011\bar{1})$ ,  $(01\bar{1}1)$  and  $(1\bar{1}01)$ .

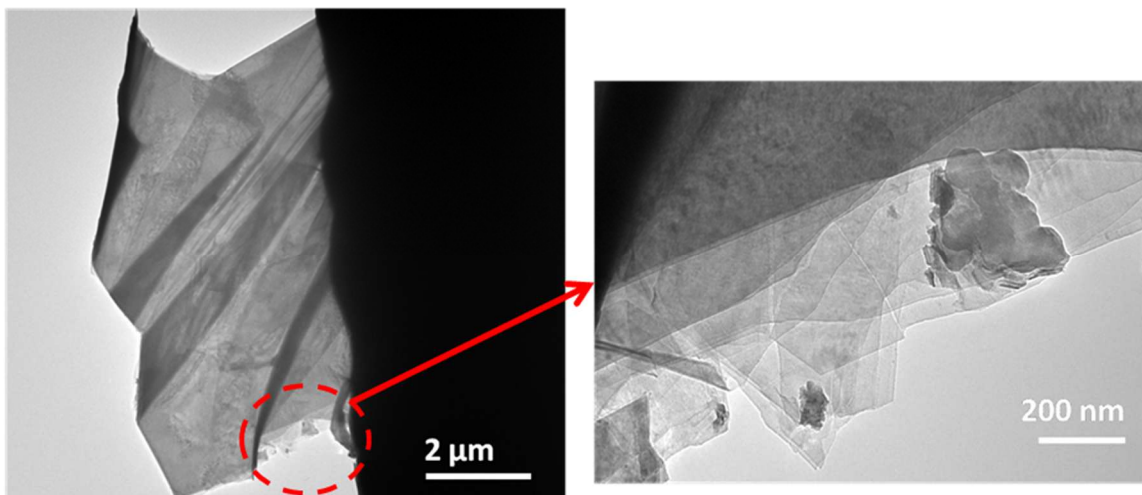
Shown in part b is the expected reciprocal lattice for a single crystal (b1), single nanosheets (b2), and stack turbostratic nanosheets (b3). The spots observed in a single crystal are broadened due to the finite size effects of the nanosheets. Part c is a schematic of a grazing scan in progress, with  $\phi_i - \phi_f$  representing the rods seen in Fig 3c of the main text. The rotation of the sample in the  $\phi$  axis causes rotation of the



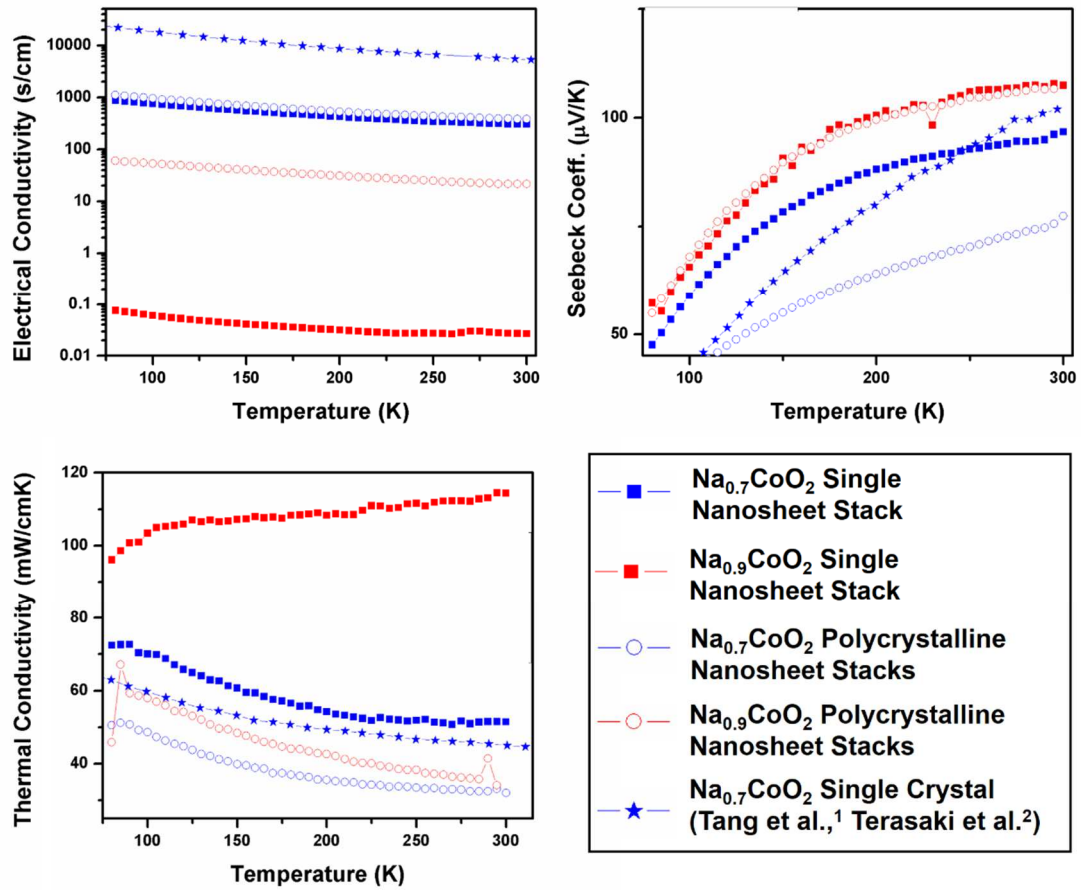
linear detector relative to the reciprocal space. As a result, a  $\theta$ - $2\theta$  type of scan is obtained along the linear detector (along  $\Delta$ , which corresponds to  $[001]$ ) for each  $\phi$  value as shown in the figure. Therefore each column of pixels of  $\phi$  in maintext **Figure 2.3c** is analogous to a scan represented by arrows in the figure above. Thus the resulting broadening corresponds to finite size along  $[001]$ .



**Figure A.7:** Optical microscope image taken in reflected (a) and transmitted (b) white light of a large exfoliated nanosheets. The features on the images bellow correlate well with those observed in TEM (**Figure 2.4a** main text).



**Figure A.8: TEM images showing single layers on the edge of an exfoliated nanosheets.** This provides a useful method of measuring the thickness of the exfoliated piece.

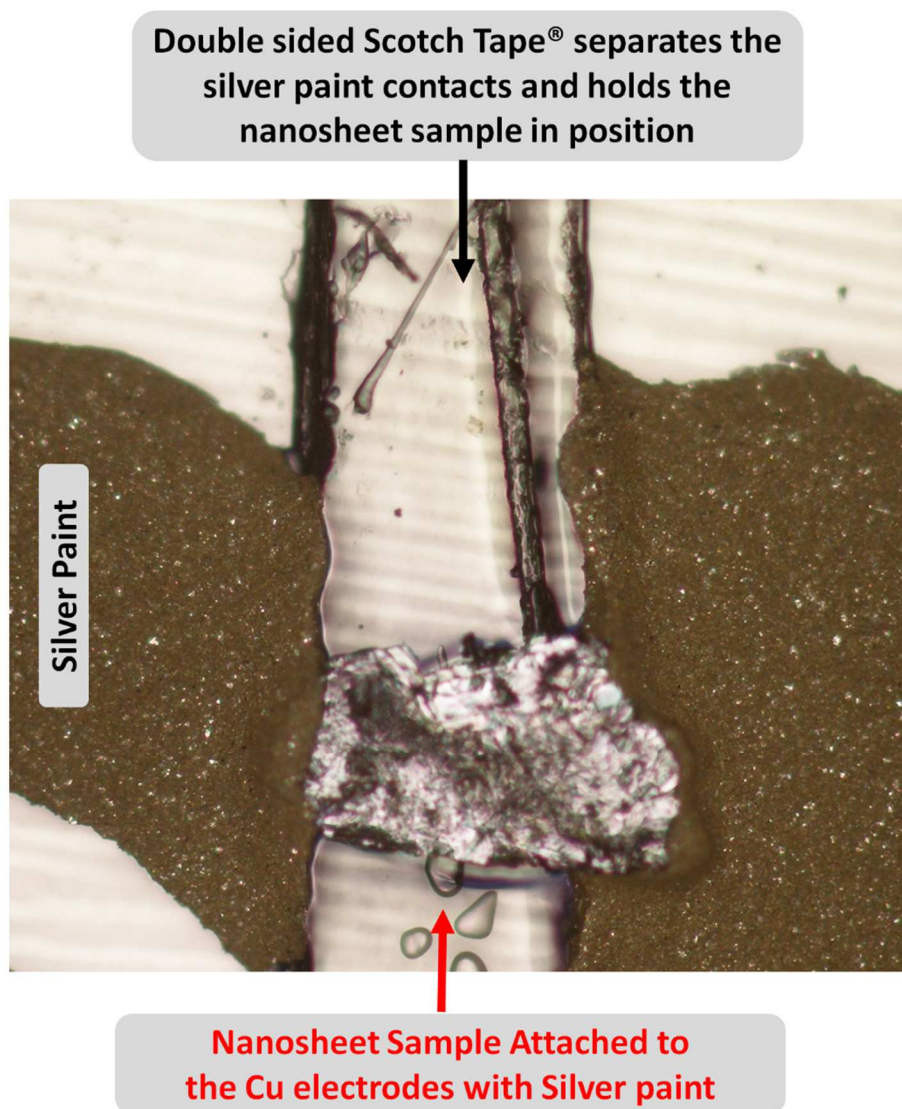


**Figure A.9: Preliminary thermoelectric measurements (electrical conductivity, Seebeck coefficient and thermal conductivity) of Na<sub>0.7</sub>CoO<sub>2</sub> (blue data points) and Na<sub>0.9</sub>CoO<sub>2</sub> (red data points) nanosheet stacks. Solid square data points are for measurements of single nanosheet stacks. For single nanosheet stacks the measurements are performed perpendicular to stacking axis. Open circle data points are for multiple nanosheet stacks pressed together in a rectangular pellet. These pellets are polycrystalline along the measurement axis. Literature data for single crystalline Na<sub>0.7</sub>CoO<sub>2</sub> thermoelectric measurements are also included for comparison (star shaped data points). Thermal conductivity of the Na<sub>0.7</sub>CoO<sub>2</sub> nanosheets is not lower than that of single crystalline Na<sub>0.7</sub>CoO<sub>2</sub>. This is probably due to perfect registry of the**

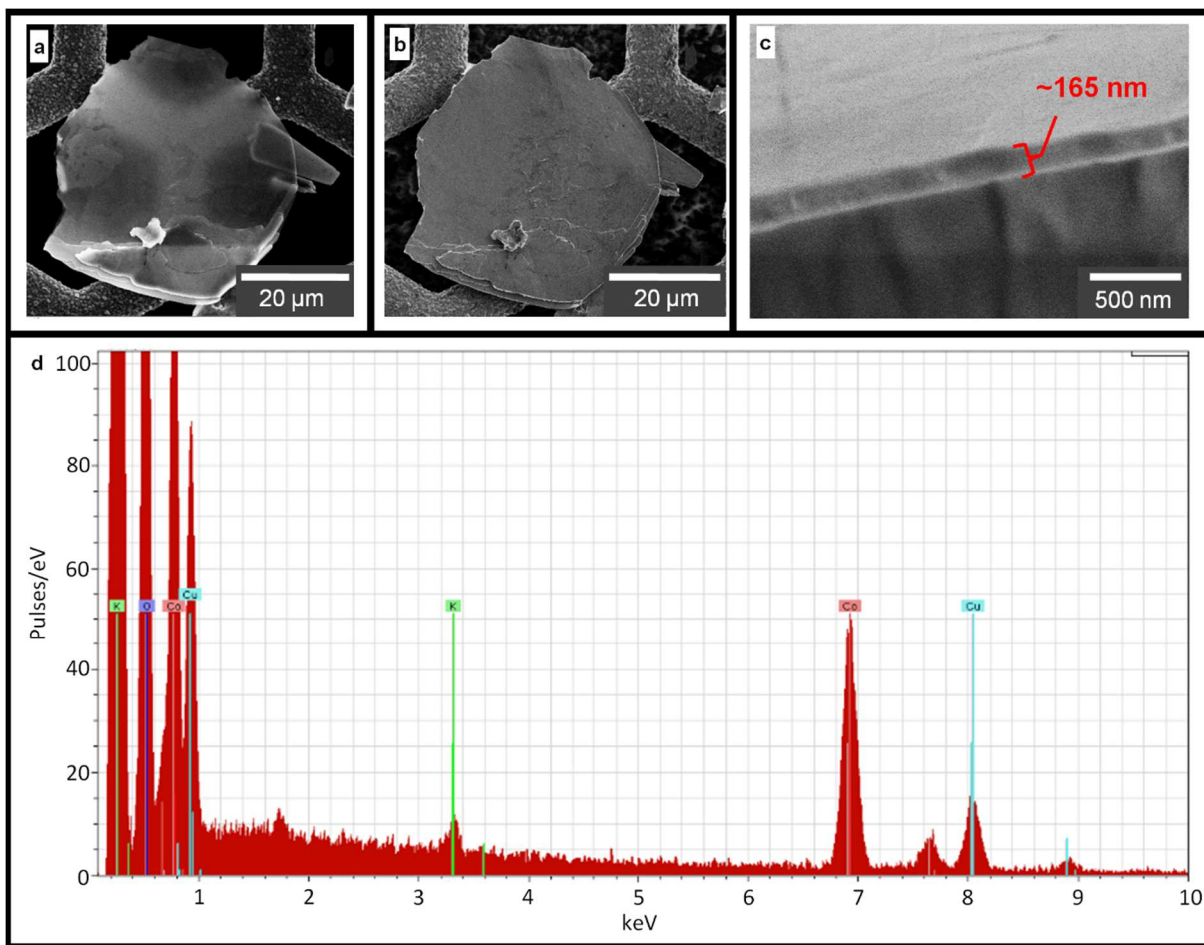
$\text{Na}_{0.7}\text{CoO}_2$  nanosheets along their c-axes which does not cause scattering of phonons. Electrical conductivity of  $\text{Na}_{0.7}\text{CoO}_2$  nanosheets is also lower compared to electrical conductivity of bulk  $\text{Na}_{0.7}\text{CoO}_2$ . This can be due to voids and cracks in the measured nanosheet stacks.

## APPENDIX B

### B. SUPPLEMENTARY INFORMATION FOR CHAPTER 3

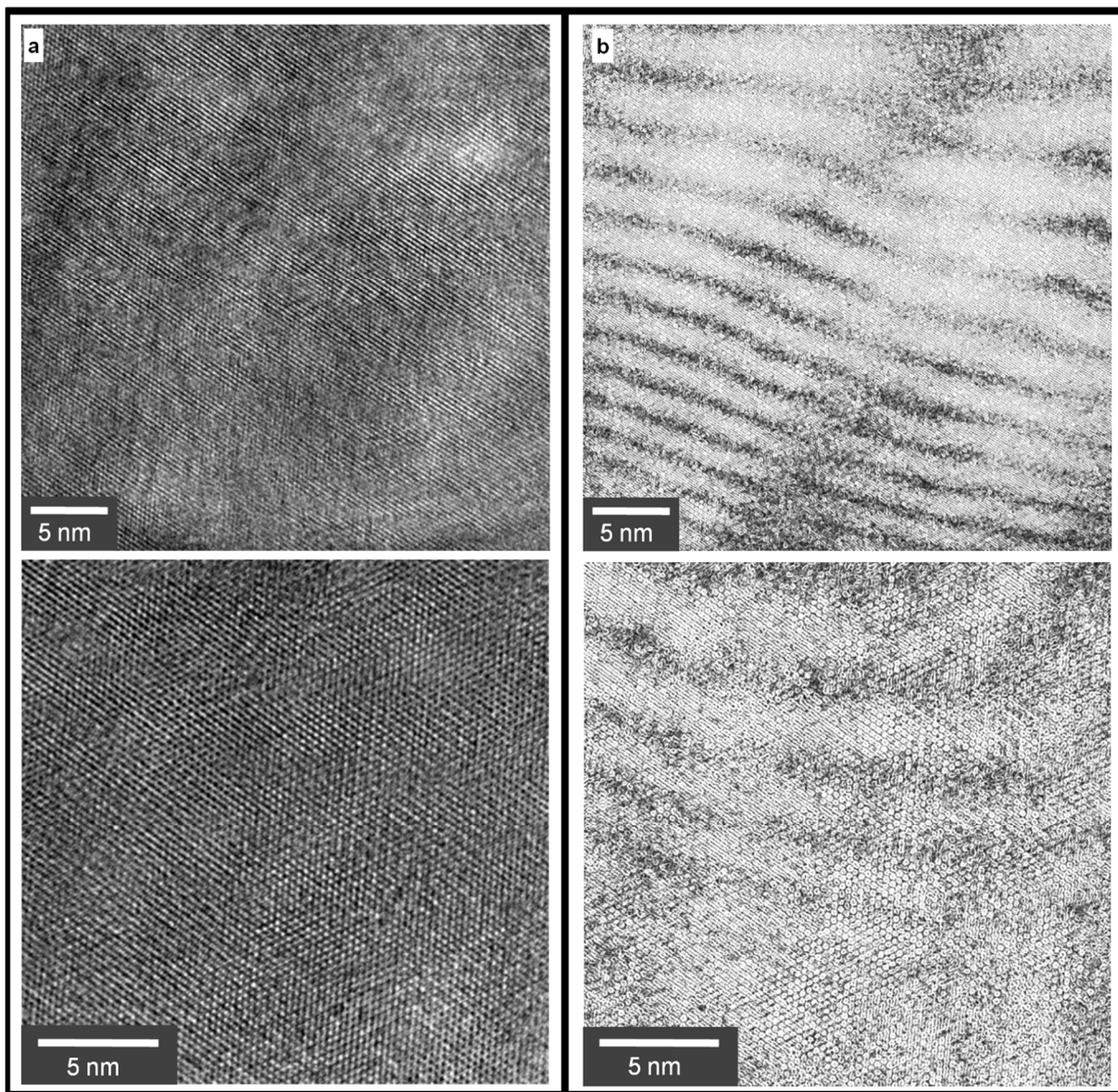


**Figure B.1:** Top view optical image of the configuration for the two-point resistivity measurements of  $K_x\text{CoO}_2 \cdot y\text{H}_2\text{O}$  nanosheet stacks. The nanosheet stack is connected for a 4-wire resistance measurement by using each of the two silver paint area contacts for both the voltage and current measurements. Note: only the top surface of the nanosheet stack can be seen in the image.

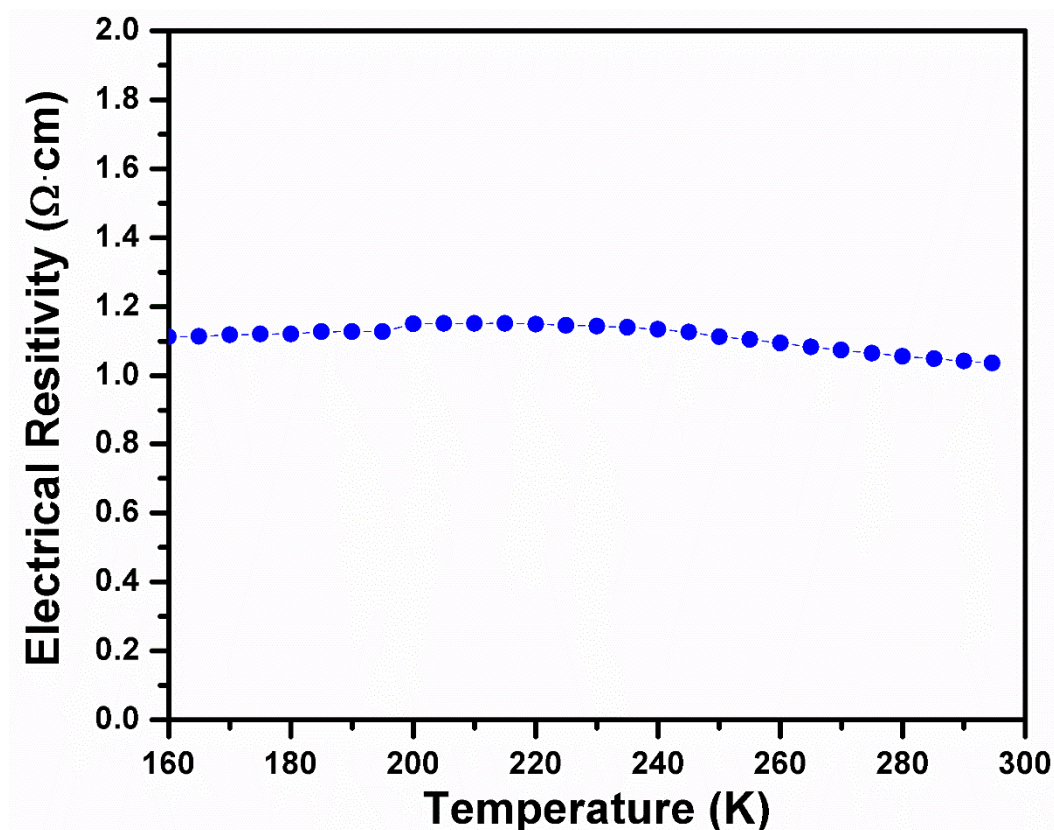


**Figure B.2: SEM images and EDS of the exfoliated  $K_xCoO_2 \cdot yH_2O$  nanosheet in Figure 3.1c (main text)** (a) SEM image taken at accelerating voltage of 15 kV. (b) SEM image taken at accelerating voltage of 2 kV. (c) SEM image of the edge of the nanosheet. The nanosheet is tilted by  $77^\circ$ . Based on the measured thickness ( $\sim 165$  nm) in the tilted image, the nanosheet is expected to be  $\sim 170$  nm thick. (d) EDS of the nanosheet taken accelerating voltage of 15 kV. EDS shows peaks for K, Co, O and Cu. The Cu peak is likely due to the TEM grid under the nanosheet.



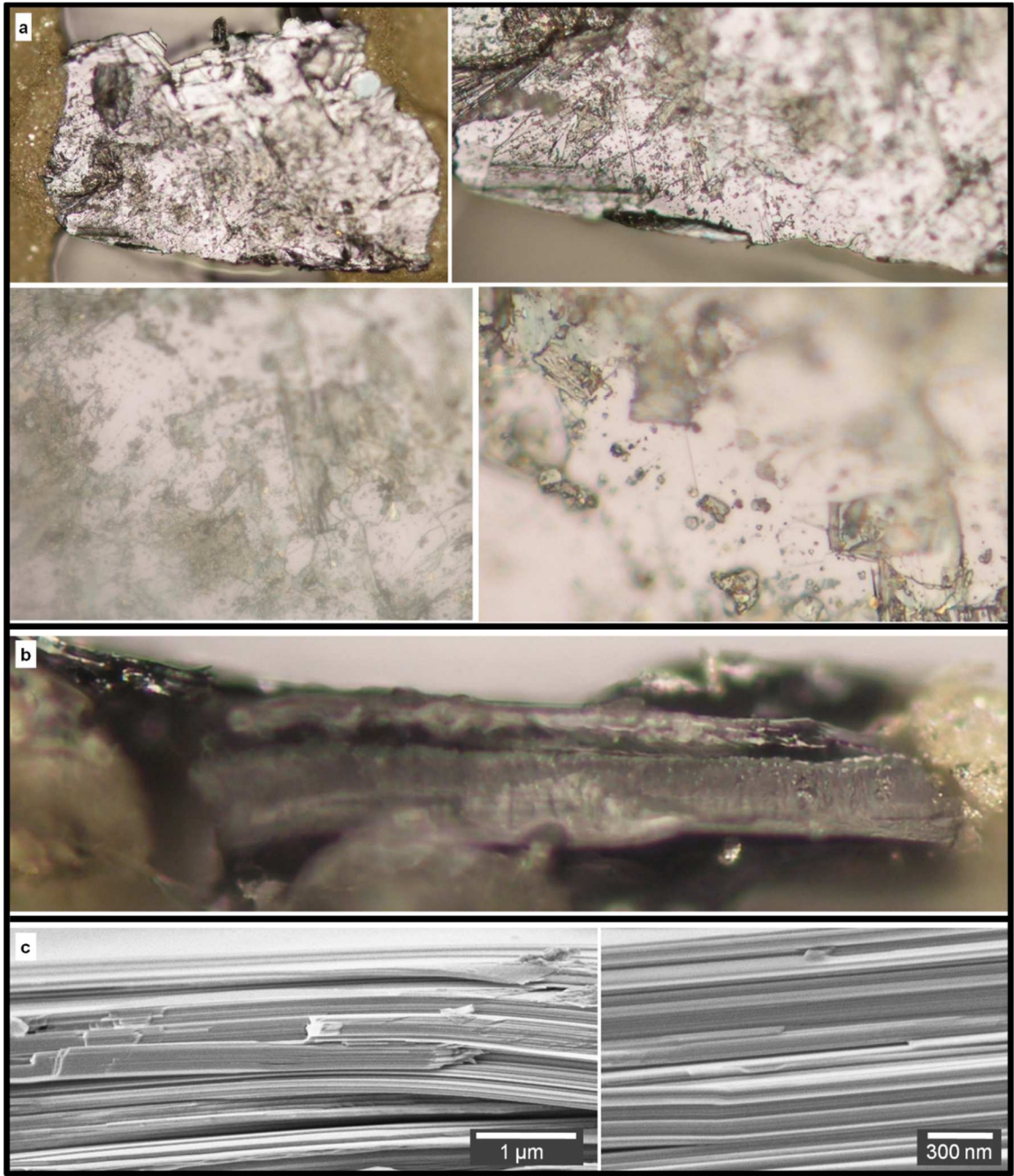


**Figure B.3: (a) HR-TEM images of two different exfoliated  $K_xCoO_2 \cdot yH_2O$  nanosheets at different magnifications. (b) Moire fringes in (a) revealed using Edge Finder function of ImageJ software**



**Figure B.4: Electrical resistivity measurement of polycrystalline  $K_xCoO_2 \cdot yH_2O$  nanosheet stacks.** In order to obtain the polycrystalline sample mildly ground water stabilized  $K_xCoO_2 \cdot yH_2O$  nanosheet stacks were uni-axially pressed into a pellet with a rectangular die set at  $\sim 250$  MPa. XRD results indicate that the nanosheet stacks maintained their c axis lattice spacing after grounding and pressure application (See **Figure B.11**). The size of the pressurized pellet is  $10.28 \times 6.25 \times 1.1$  mm (LxWxT). The relative density of the polycrystalline pellet is estimated to be  $\sim 70\%$ . Electrical resistivity measurements were performed perpendicular to the pressure application axis. Electrical resistivity of the polycrystalline sample is  $>200$  times higher than the single nanosheet stack (See maintext **Figure 3.2**), likely due to the grain boundary resistivity and significantly higher cross-plane resistivity<sup>1</sup>.

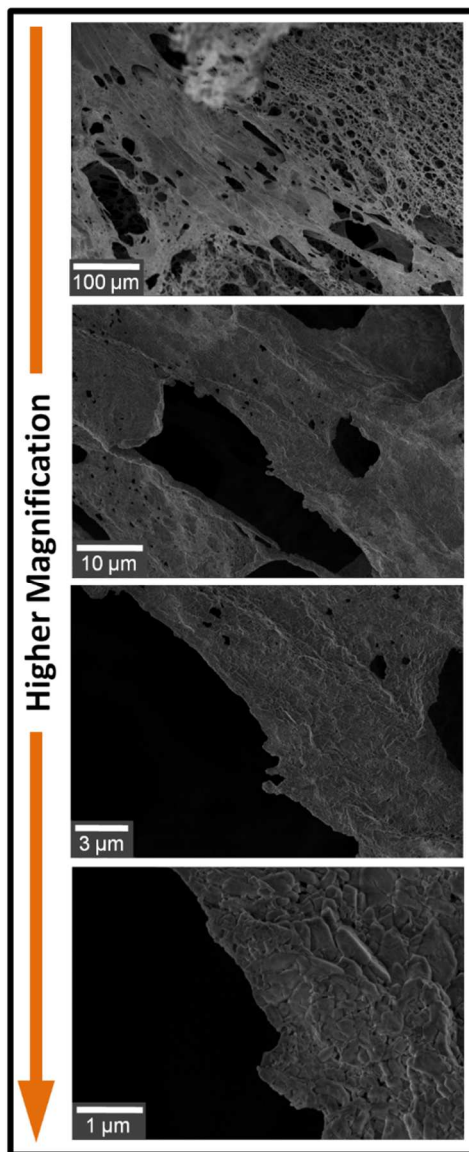




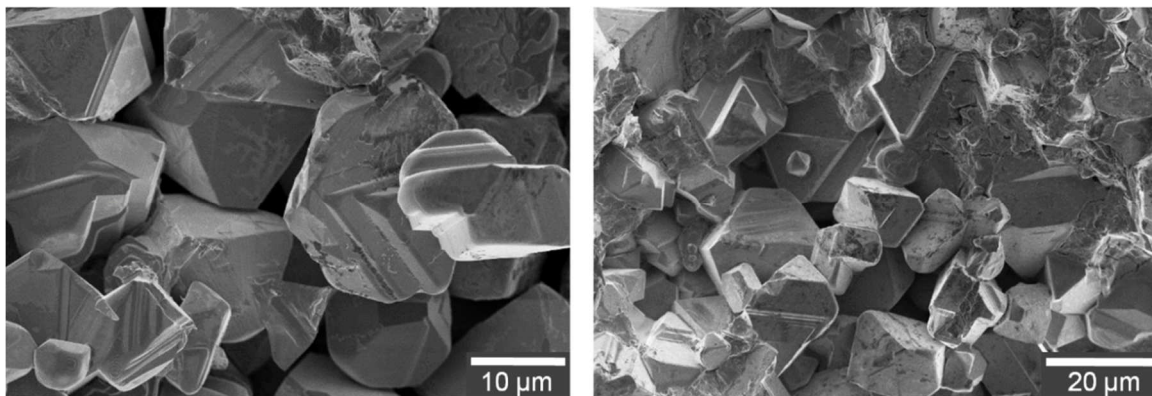
**Figure B.5: Optical microscope images and SEM images of the sample tested for electrical resistivity measurements in Figure 3.2 of the maintext (see also Figure B.1 for the image of the resistivity measurement setup). (a) Low to high magnification optical microscopy images of the sample surface. Although the surface has excessive**

dents and scratches due to mechanical handling during the extraction of the nanosheet stack from the calcined pellet and the preparation of the resistivity measurement setup, the images show the reflective nature of the single crystalline nanosheet top surface.

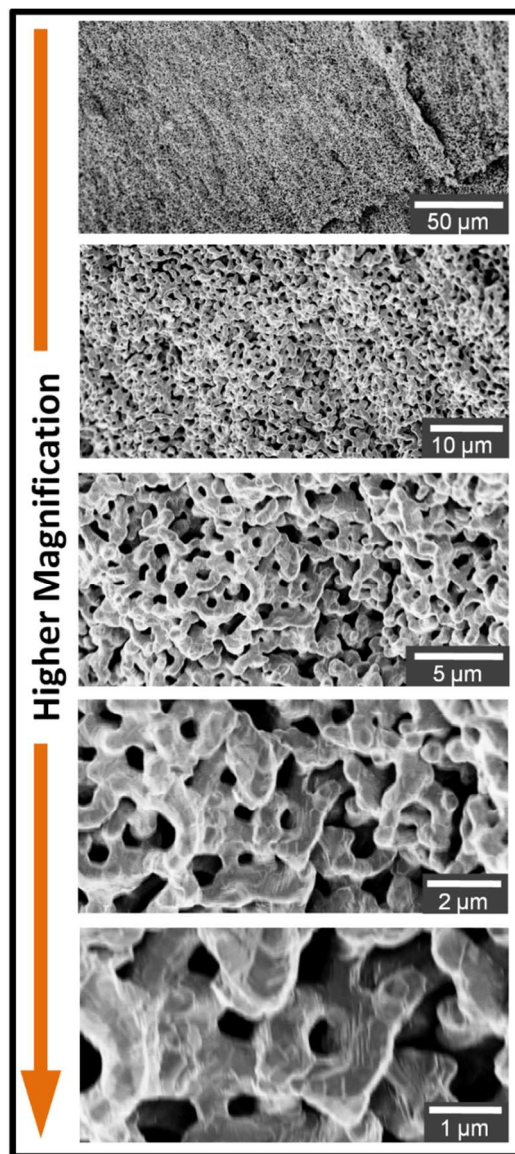
(b) Optical microscopy image of the nanosheet stack from the edge. Unlike the nanosheet stack top surface, the edge of the nanosheet stack is not reflective probably due to roughness introduced by the lamellar structure. To image this sample from edge it was necessary to detach the sample from the substrate by peeling off the scotch tape under the sample from the substrate. The mechanical force applied during this process partially damaged and cracked the single nanosheet stack sample along the measurement plane, resulting in two separate stacks of nanosheet (as seen in optical image). Cracking along this axis is preferred since the nanosheets are stacked and the measurement plane is perpendicular to the nanosheet stacking axis. (c) High magnification SEM images of the nanosheet stacking in the measured sample. Due to charging problems the sample had to be detached from the scotch tape which resulted in the sample to break into small pieces. Therefore low magnification SEM images of the entire nanosheet stack cannot be provided.



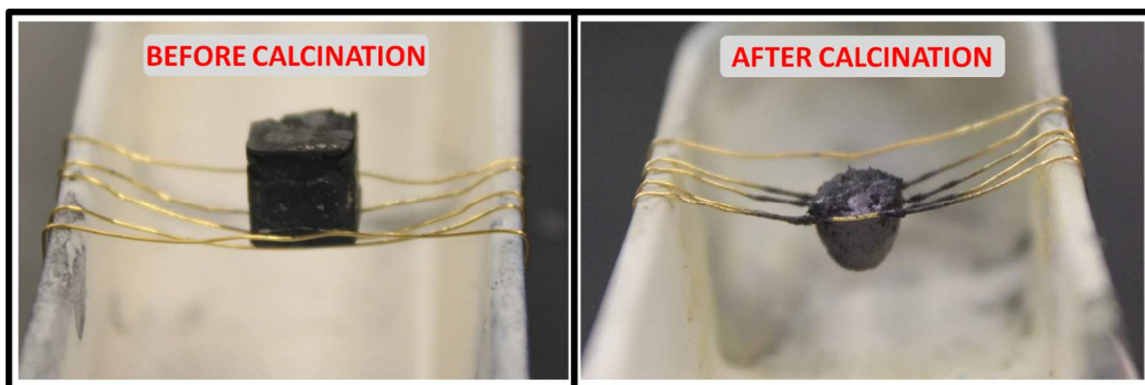
**Figure B.6: Low (top) to high (bottom) magnification SEM images of the flakes that appear after the autocombustion of the PAA-Metal Complex aqueous solution.** The thickness of the flakes is  $< 200$  nm and the length measures up to 400  $\mu\text{m}$ .



**Figure B.7: Samples with identical processing conditions but without kinetic demixing.** Nanosheets are not formed without kinetic demixing and particle size is smaller than  $\sim 30\text{ }\mu\text{m}$ .

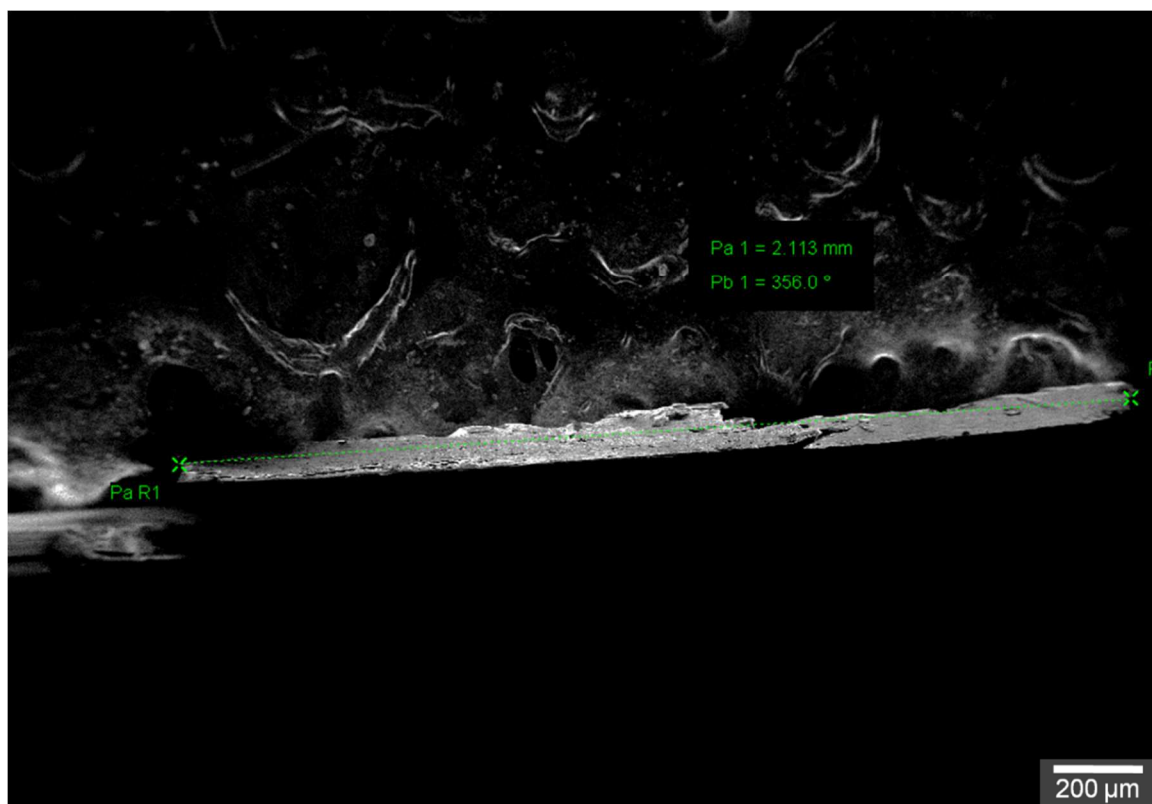


**Figure B.8: Low (top) to high (bottom) magnification SEM images of the K-deficient region of the pressurized pellet after the e-field induced kinetic demixing process.** The pellet was a homogenous mixture of K, Co and O atoms prior to the e-field induced kinetic demixing process.



**Figure B.9: Optical image of the K-rich portion of the kinetically demixed pellet before and after calcination.** In the “after calcination” image (right) the sample appears as a droplet in a semi-spherical form, indicating molten fluidity occurred during the calcination. Due to the molten fluidity, the K-rich pellet penetrates through the Au wires used for suspending the sample.





**Figure B.10: Low magnification SEM image of a ~2.1 mm long  $K_xCoO_2 \cdot yH_2O$  nanosheet stack**

### ***B.0. Supplementary Methods: Area Detector XRD Sample Mounting***

Both the single nanosheet stack and polycrystalline nanosheet samples are mounted on sample stage using Scotch Tape. The polycrystalline nanosheet sample is oriented on the sample stage such that the pressure application axis for the pellet formation is perpendicular to the sample stage. Since the single nanosheet stack sample is slightly smaller than the x-ray beam spot size the x-ray beam is also expected to hit the sample stage. In order to minimize x-ray scattering and diffraction from the sample stage a single crystalline Si wafer is placed between the single nanosheet stack sample and the sample stage. Although the incoming x-ray beam partially hits Si wafer, no Si peaks are observed since the diffraction condition is not met for the lattice planes of the single crystalline Si wafer.

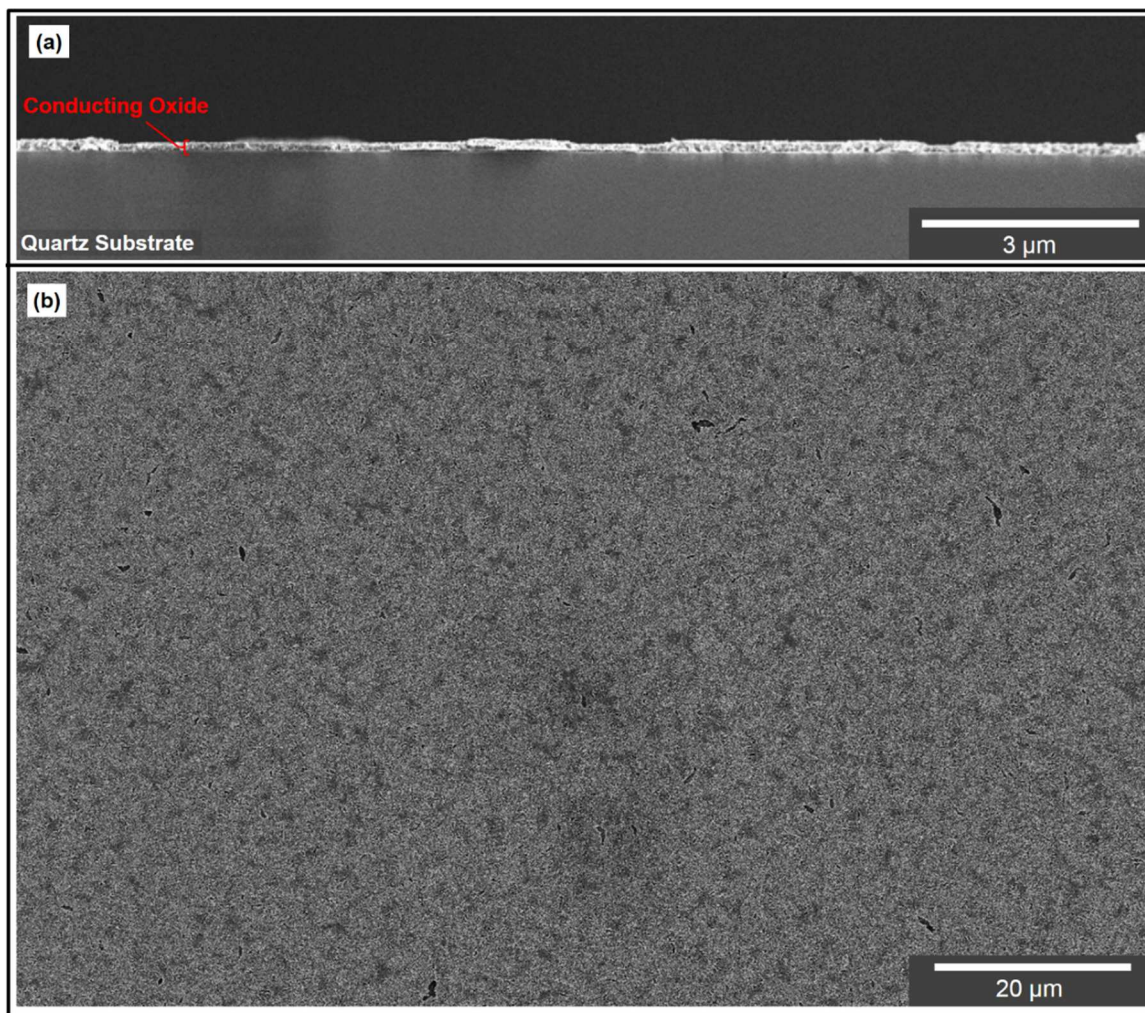
### **REFERENCE(S)**

- <sup>1</sup> H. Y. Tang, H. Y. Lin, M. J. Wang, M. Y. Liao, J. L. Liu, F. C. Hsu, and M. K. Wu, *Chem Mater* **17** (8), 2162 (2005).



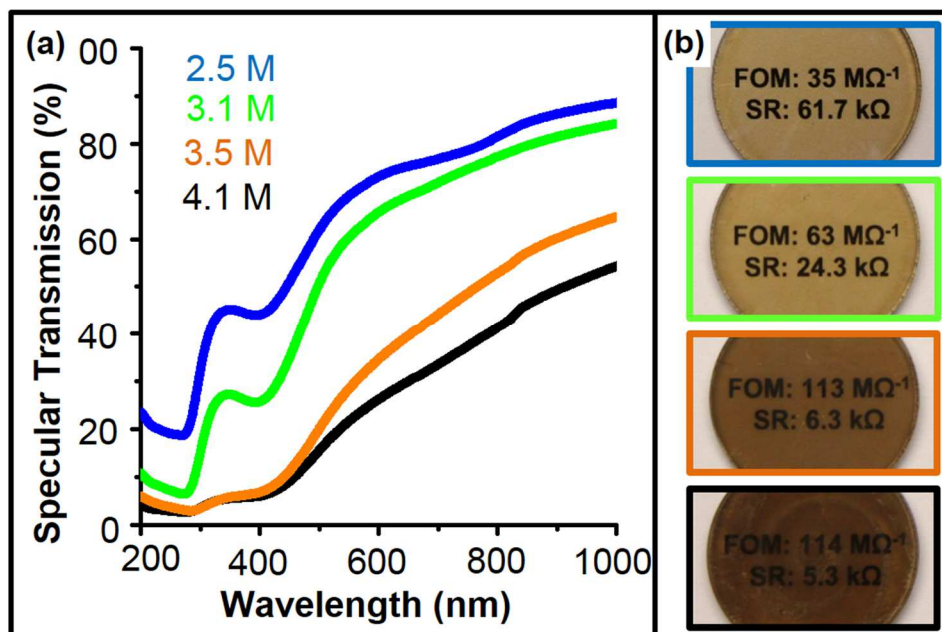
## APPENDIX C

### C. SUPPLEMENTARY INFORMATION FOR CHAPTER 4



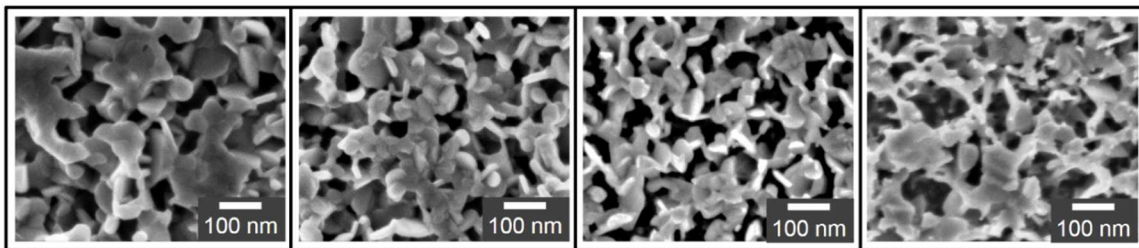
**Figure C.1: Low magnification SEM images of the TCO thin film in the maintext**

**Figure 4.1.** (a) Low magnification cross-sectional image of the TCO thin film showing long range uniformity. (b) Low magnification surface image of the TCO thin film showing the long range smoothness and occasional micron-scale cracks.

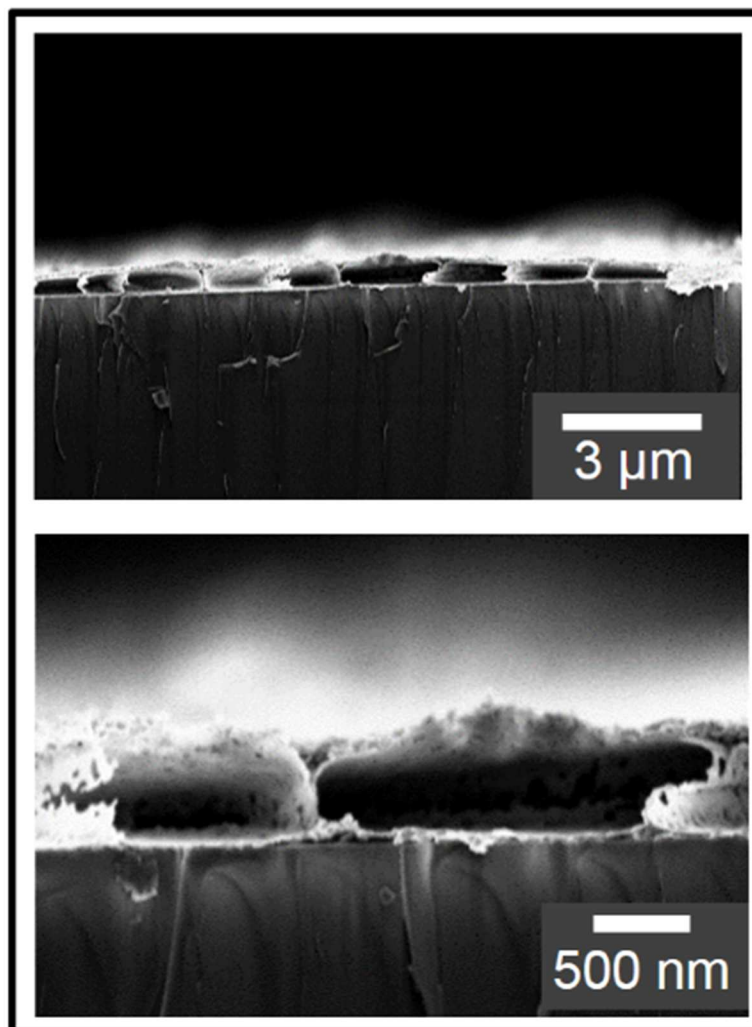


**Figure C.2: Transparent conductive oxide (TCO) thin films of p-type  $\text{Ca}_3\text{Co}_4\text{O}_9$  obtained from viscous resin intermediates with different solute concentrations.**

All of the TCO films in this figure are calcined at 650 °C for 15 minutes. Different solute concentrations of 2.5 M, 3.1 M, 3.5 M and 4.1 M are indicated with blue, green, orange and black colors respectively. **(a)** Optical transmission measurements of the TCO thin films. **(b)** Camera images showing the  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin films in **(a)** on white paper with FOM and sheet resistance (SR) values written under the samples.

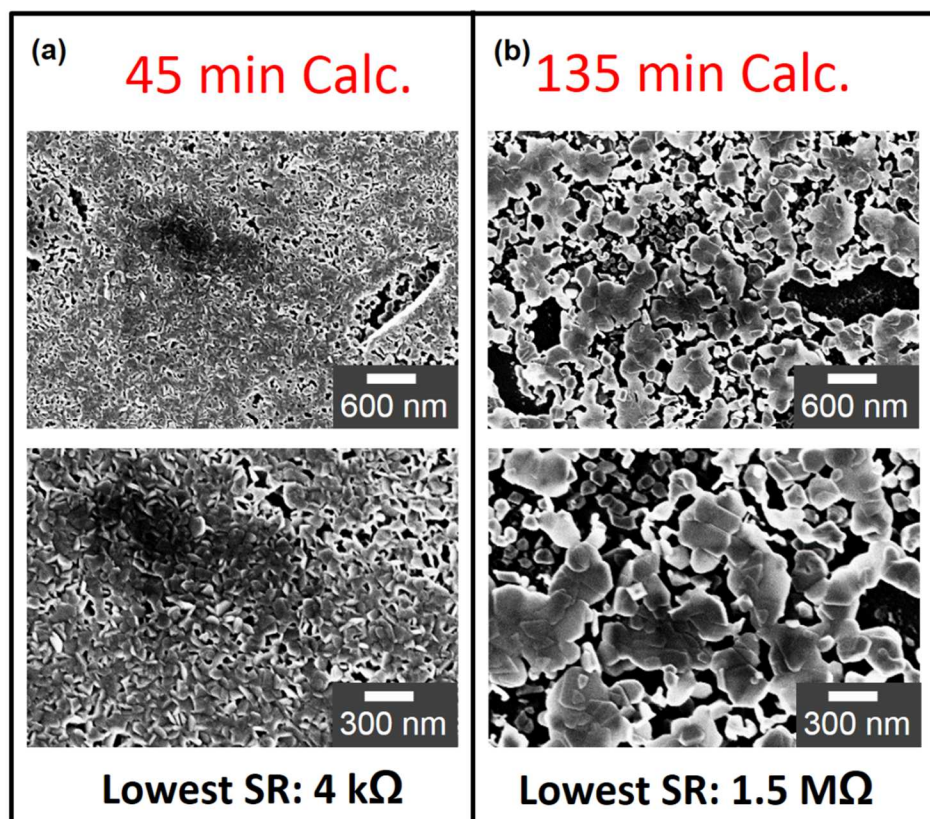


**Figure C.3: Surface SEM images of TCO films made from resins with different solute concentrations.** All of the films are calcined at 650 °C for 25 minutes. The thin films are made from resins with solute concentrations of **(a)** 4.1 M, **(b)** 3.5 M, **(c)** 3.1 M and **(d)** 2.5 M.



**Figure C.4:** Cross-sectional SEM images of the  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin film prepared from 4.1 M solute concentration resin after 15 minutes calcination at 650 °C.





**Figure C.5:** SEM images of two  $\text{Ca}_3\text{Co}_4\text{O}_9$  thin films that are calcined for (a) 45 and (b) 135 minutes at 650 °C. Both films are prepared from same solute concentration resin. The lowest sheet resistance values measured on the films are 4 kΩ and 1.5 MΩ respectively for (a) and (b).

## APPENDIX D

### D. SUPPLEMENTARY INFORMATION FOR CHAPTER 5

#### ***D.0. Supplementary Discussion: Casimir-Ziman Surface Scattering Regimes***

From Eq. 5.1 and Eq. 5.2 of the main text, we can see that a key characteristic of Casimir-limit scattering behavior (i.e. when  $l_{eff} = l_0$ ) is that  $l_{eff}$  is independent of phonon frequency. For surface roughness  $\eta$ , we can distinguish three distinct regimes of scattering behavior.

First, wavelengths much greater than  $\eta$  will reflect specularly, allowing the phonon to travel unimpeded down the nanosheet. Then  $l_{eff}$  is infinite, and transmission through the nanosheet should be 100%, independent of nanosheet length.

Secondly, in the opposite limit, wavelengths much smaller than  $\eta$  will scatter diffusively every time they strike the surface, emerging in a random direction, which reduces  $l_{eff}$  to equal  $l_0$  (the so-called 'Casimir limit' value). For sheets of rectangular cross-section, analytic expressions for  $l_0$  have been determined by McCurdy et al and have a value slightly larger than the width of the sheet.<sup>1</sup> In the 'Casimir limit', transmission through the nanosheet will be independent of wavelength, but will decrease for increasing sheet length.

Thirdly, in an intermediate regime, wavelengths  $\lambda \approx \eta$  will be partially scattered diffusively and partially specularly reflected. In this regime, transmission will depend on both wavelength and channel length.

### D.1. Supplementary Discussion : Detailed Explanation of Phonon Pathways

Here we reproduce Eq. 5.4 from the main text and describe its components in more detail. The rate  $\dot{n}_{ph,d}$  of phonons striking the detector may comprise four possible components, where  $s$  (source),  $d$  (detector) and  $ns$  (nanosheets) indicate the phonon pathways (**Figure 5.2e** of main text):

$$\begin{aligned}
 \dot{n}_{los}(s \rightarrow d) & \quad (\text{ballistic through bulk Si, frequency-independent}) \\
 \dot{n}_{bs} & \quad (\text{scattering from substrate, possibly frequency-dependent}) \\
 \dot{n}_{los}(s \rightarrow ns \rightarrow d) & \quad (\text{ballistic through nanosheets, frequency-independent}) \\
 \dot{n}_{sc}(s \rightarrow ns \rightarrow d) & \quad (\text{scattering \& reflection through nanosheets, frequency-dependent})
 \end{aligned}$$

Eq. D.1

First, in regions where nanosheets are absent, phonons travel ballistically in direct line-of-sight from source to detector, with phonon arrival rate  $\dot{n}_{los}(s \rightarrow d)$ . This rate will be independent of phonon frequency  $\omega$  and will depend on the solid angle  $\Omega_d$  defined by the detector relative to source STJ.

Second, a rate of phonons  $\dot{n}_{bs}$  may strike the detector and its attached wiring leads after back-scattering from the bottom surface of the chip or from resonant-scattering impurities within the silicon.<sup>2,3</sup> This rate may have some dependence on phonon frequency  $\omega$  but should depend primarily on the scattering probability on the back-side of the Si chip; similar spectrometers on the same chip thus should register the same level of backscatter  $\dot{n}_{bs}$ , regardless of  $\Omega_d$  or of whether nanosheets are

present.

Third, if nanosheets occupy the line-of-sight path, then a reduced portion of ballistic line-of-sight flux  $\dot{n}_{los}(s \rightarrow ns \rightarrow d)$  may transit to the detector after traversing the nanosheets without striking their walls. This rate is not expected to depend on  $\omega$ .

Fourth, a rate of phonons  $\dot{n}_{sc}(s \rightarrow ns \rightarrow d)$  reaches the detector after scattering or reflecting within the nanosheets. This will depend on the nanosheet geometry as well as the source and detector geometries, and is expected to be strongly frequency-dependent. This component is what allows us to probe Casimir-Ziman scattering behavior.



## ***D.2. Supplementary Discussion: Description of Measurement of Background Level***

$\dot{n}_{bs}$

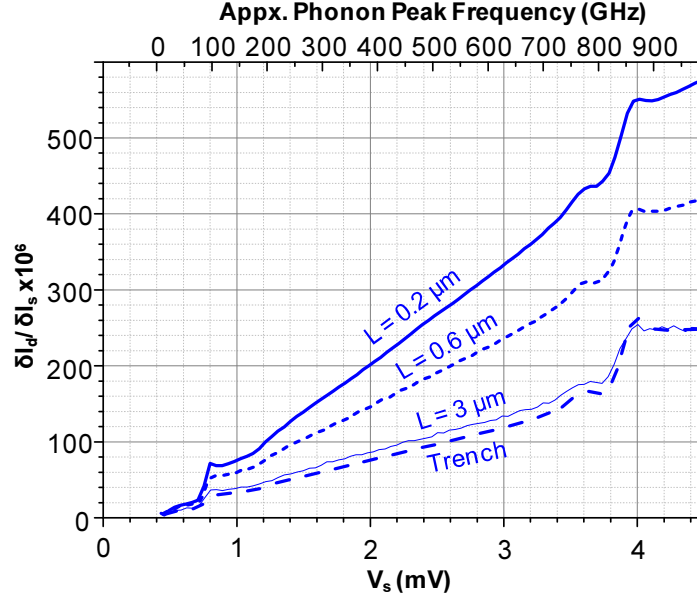
As described in Eq. 5.6 of the main text, if nanosheets are present, we expect the phonon arrival rate at the detector to be

$$\dot{n}_{ph,a}(\omega) = \dot{n}_{los}(s \rightarrow ns \rightarrow d) + \dot{n}_{sc}(s \rightarrow ns \rightarrow d) + \dot{n}_{bs} + \dot{n}_{los}(s \rightarrow d)$$

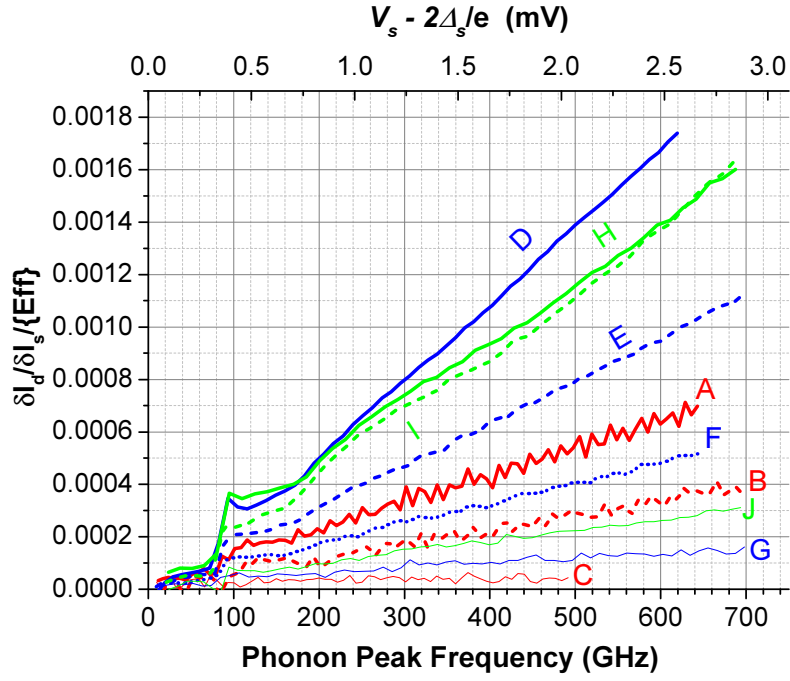
Eq. D.2

To assess the scattering transmission behavior, we must distinguish the backscatter signal  $\dot{n}_{bs}$  from the other components; this level may be comparable in size or even larger than the other two signal components (**Figure D.1**, Appendix D). Since we expect that  $\dot{n}_{bs}$  is likely to be the same for all spectrometers on the same sample chip (see preceding discussion), a convenient way to measure  $\dot{n}_{bs}$  is through a spectrometer in which a trench is etched into the mesa to completely block the line-of-sight path for phonons, but is otherwise identical to the other spectrometers on the chip.<sup>4</sup> Where this was not possible, we estimate  $\dot{n}_{bs}(\omega)$  instead by measuring  $\delta I_d / \delta I_s$  of a phonon source and detector that are separated by  $\sim 100 \mu\text{m}$  on the same chip and do not share a line-of-sight. For such a source/detector pair, the trajectory for scattering from the bottom of the chip ( $\sim 500 \mu\text{m}$  away) is much the same as for a more closely separated source/detector pair. The resulting signal level is slightly lower than that found in a ‘trench’ spectrometer. Comparing several such measurements of different pairs of sources and detectors on a single chip, we observe variations of  $\sim 10$  to 20% in the backscatter signal level. These variations may be attributable to

different scattering rates in different regions of the chip backside. We use these variations to estimate the systematic error in the subtraction of the level  $\dot{n}_{bs}(\omega)$ .



**Figure D.1: Phonon transmission measurement through three different types of nanosheet array on the same sample chip.** For comparison, transmission through a spectrometer in which a trench blocks the line-of-sight path between source and detector is also shown. This ‘trench’ spectrometer is located on the same chip as the other measurements in this plot, and constitutes the measured  $\dot{n}_{bs}$  signal level on this chip. The spectrometers in this figure are samples D, E and G (see **Table 5.1** in main text.) The transmission signal levels in this plot were subsequently corrected for detector efficiencies and had the ‘trench’ spectrum subtracted to remove  $\dot{n}_{bs}$ , and appear as part of **Figure D.2**, Appendix D, below and as part of **Figure 5.3** in the main text.



**Figure D.2: Phonon transmission measurement through ten different types of nanosheet array A through J. (See Table 5.1 and Table 5.2 in the main text.)**

Signals are corrected for estimated efficiency of each detector, and estimated background level  $\bar{n}_{bs}$  has been subtracted from each measurement trace. From this plot we select the signal values for phonon peak frequency of 400 GHz, and compare these values against calculated phonon transmission factors, as shown in **Figure D.8** and in **Figure 5.6** of the main text.

### ***D.3. Supplementary Discussion: Additional Details of Monte Carlo Calculations of Phonon Focus Behavior***

The directional distribution of phonon group velocity vectors are significantly affected by the “phonon focusing” phenomenon.<sup>5,6</sup> Incorporation of phonon focusing factors is thus essential for accurate modeling of phonon transmission in either the analytical or MC models. Because of the elastic anisotropy of crystalline transport media, phonon energy concentrates along particular crystallographic directions and depletes them along other directions. We incorporate phonon focusing into the MC model as follows. A phonon emerging from the source is assigned a random direction for its  $k$  vector. We then solve the non-dispersive Christoffel equation for a cubic crystal. The material parameters (elastic constants and density) required for this equation are listed in **Table D.2**. The resulting group velocity vector indicates the direction of travel of this phonon.<sup>7,8</sup> The algorithm is repeated at least  $10^6$  times, a large enough number of times to cover all  $k$  vector directions and establish good statistics. We do an approximately equal number of repetitions for each of the three phonon polarizations: LA, ST and FT. This calculation is undertaken only for phonons emerging from the source, and should accurately account for the phonon focusing effects on ballistic line-of-sight phonon propagation as well as specular reflection. If a phonon scatters diffusively from nanosheet sidewalls, we assume its subsequent propagation to be isotropic, as we assume that the randomization associated with diffusive scattering will wash out the phonon focus behavior. Incorporation of the phonon focusing factors in the analytical approach requires a 2D phonon focusing

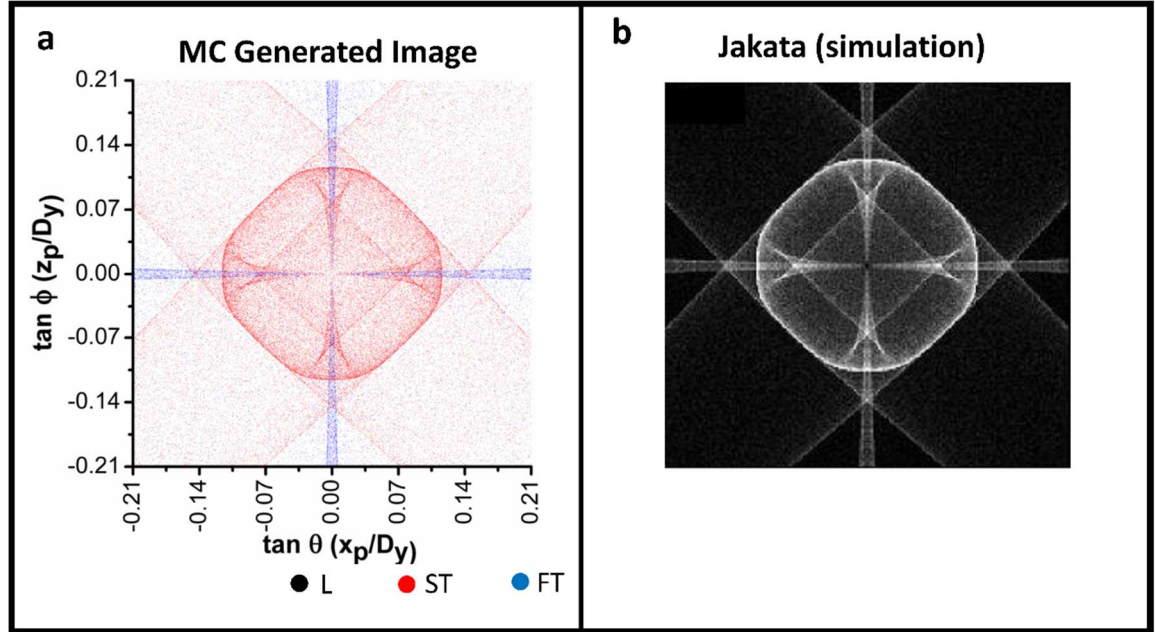
factor map of the Si (110) plane, which can be generated by the MC simulations.

**Figure 5.4** of the main text displays the angular 2D phonon focusing factor map on the (110) plane of Si.

(See Appendix D.6 for C++ source code of the Monte Carlo model)

#### ***D.4. Supplementary Methods: Generation of 2D phonon focusing factor map for angles relative to (110) direction, using the Montel Carlo Approach***

In the MC simulations for generating the 2D phonon focusing factor map, a detector plane perpendicular to [110] (y axis) is positioned at a distance ( $\Delta y$ ) from a point phonon source such that the detector plane is centered at  $x = z = 0$ . The detector plane is divided into 1000 by 1000 bins of equal area along  $x$  and  $z$  directions. The MC model is run with and without the phonon focusing algorithm and the number of phonons collected by the bins are recorded. The ratio of the number of phonons collected by a bin with and without the phonon focusing effect gives the phonon focusing factor  $A_{foc}(x_n, z_n)$  for that bin. The  $x$  and  $z$  coordinates of the bins are converted into angular coordinates  $\varphi_n = \arctan(\frac{x_n}{\Delta y})$  and  $\theta_n = \arctan(\frac{z_n}{\Delta y})$  where  $n$  is the index number for the bins.



**Figure D.3: Phonon focusing image on (100) plane of Si, generated using MC approach** (a). The MC generated phonon focusing image matches with images previously reported by Jakata through simulations<sup>8,9</sup> (b) and Tamura et. al. through experiment and simulations (Figure 3a and 3b respectively in Ref. 14)<sup>10</sup>. \*Simulation image in (b) is from Dr. Kudakwashe Jakata's Master's Thesis and copyrighted to University of Witwatersrand. This image is reprinted with kind permission of University of Witwatersrand.



### ***D.5. Supplementary Discussion: Comparison of Monte Carlo and Analytic Calculations of Line-of-Sight Ballistic Phonon Transmission Factor***

In **Figures 5.5** and **5.6** of the paper and related discussions, we compare measured rates of phonon transmission to a MC calculation of the transmission factor. In cases where phonon-surface scattering may be neglected, we compare these calculated transmission factors to values found using a simpler analytic expression (See Appendix D.7 for Matlab source code of the analytical model). The ballistic line-of-sight transmission factor  $\Gamma_{los}$  is defined in equation (5) of the paper for the case of bulk line-of-sight transport, and may also be applied to find line-of-sight transport through nanosheets. The algorithm for the analytical model is derived by calculating the approximate fractional solid angle subtended by the detector from the generator. The generator and detector are divided into rectangular segments. Each generator segment is considered as a point source and all segments are considered to emit an equal rate of phonons. For the  $n^{\text{th}}$  generator-detector segment pair, the fractional solid angle subtended by the detector segment from the generator segment is approximated by  $\Omega_n = \frac{A_{ds\perp}}{2\pi|\bar{R}_n|^2}$ , where  $\Omega_n$  is the fractional solid angle,  $\bar{R}_n$  is the position vector from the center point of the generator segment to the center point of the detector segment,  $|\bar{R}_n|$  denotes the magnitude of  $\bar{R}_n$  and  $A_{ds\perp}$  is the area of the detector segment projected onto a plane perpendicular to  $\bar{R}_n$ .

For a given arrangement & geometry of source STJ, detector STJ and bulk crystal or nanosheet array, the fractional solid angles were calculated for all  $m$  possible generator-detector segment pairs, and summed together to find the total line-of-sight

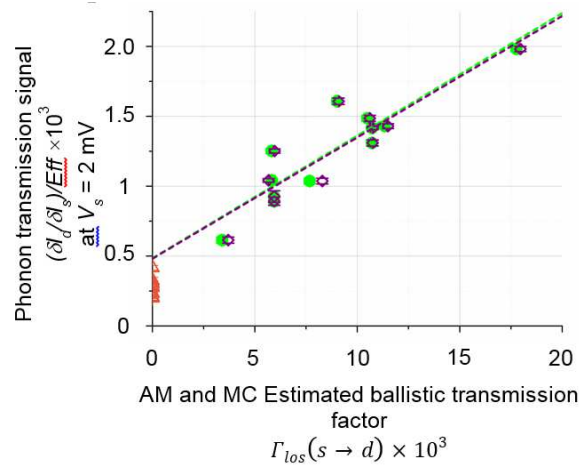
ballistic phonon transmission factor.  $\Gamma_{los} = \sum_{n=1}^m 2\cos(\theta_n) \cdot A_{foc,n} \cdot \Omega_n$ . The factor of  $2\cos(\theta_n)$  introduces a Lambertian distribution to the emitted phonon flux, where  $\theta_n$  is the angle between  $\bar{R}_n$  and the normal to the silicon surface at the STJ source. The effect of the mesa-top surface is included in this calculation of  $\Gamma_{los}$ , by assuming that this surface is specular. The sum then includes line-of-sight paths from the source to a mirror image of the detector with respect to the mesa top plane.

The phonon focusing factor  $A_{foc,n}$  for the  $n$ th source-detector segment pair is found by using the 2D angular phonon focusing factor map generated for the (110) direction of Si. (See **Figure 5.4** of the maintext.) The  $\phi$  and  $\theta$  spherical-coordinate components of  $\bar{R}_n$  are calculated, and  $\bar{R}_n$  is assigned an angular width in  $\phi$  and  $\theta$  coordinates, based on the size of the detector segment. The angular width in  $\phi$  and  $\theta$  coordinates also corresponds to a rectangular window in the 2D phonon focusing factor map. The phonon focusing factors in the map are averaged within this angular window to find  $A_{foc,n}$ .

Since  $A_{ds\perp}$  refers to a planar surface instead of a spherical surface, the exact value of the fractional angle can only be obtained for an infinitesimally small  $A_{ds\perp}$  and for more accurate analytical calculations the detector needs to be divided into smaller segments. Similarly one needs to divide the generator area into smaller segments for a more accurate representation of the homogenous phonon emission from the entire generator area.

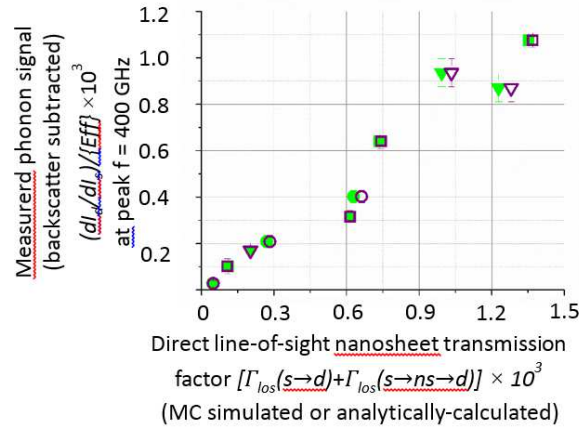
In **Figures D.4 and D.5**, we present the results of these calculations of  $\Gamma_{los}(s \rightarrow d)$  and  $\Gamma_{los}(s \rightarrow ns \rightarrow d)$ . These figures reproduce from **Figures 5.6 and 5.7** of the maintext the measured signal levels and the ballistic line-of-sight transmission

factors calculated using MC simulations. The analytically calculated transmission factors are presented on the same axes. In both cases, these results show very good agreement between the analytical calculations and MC simulations. The consistency between these two independent models confirms the validity of the MC method to model ballistic phonon propagation. The trials presented in **Figure D.6** and **Table D.4** establish the further validity of the MC algorithm to model partially-diffusive, partially-specular scattering from nanosheet surfaces. The MC approach is well-suited for modeling phonon transport through nanostructures because it can handle stochastic phenomena such as phonon-surface diffusive scattering, (Eq. (1) of paper) and it can readily handle interactions with multiple surfaces.



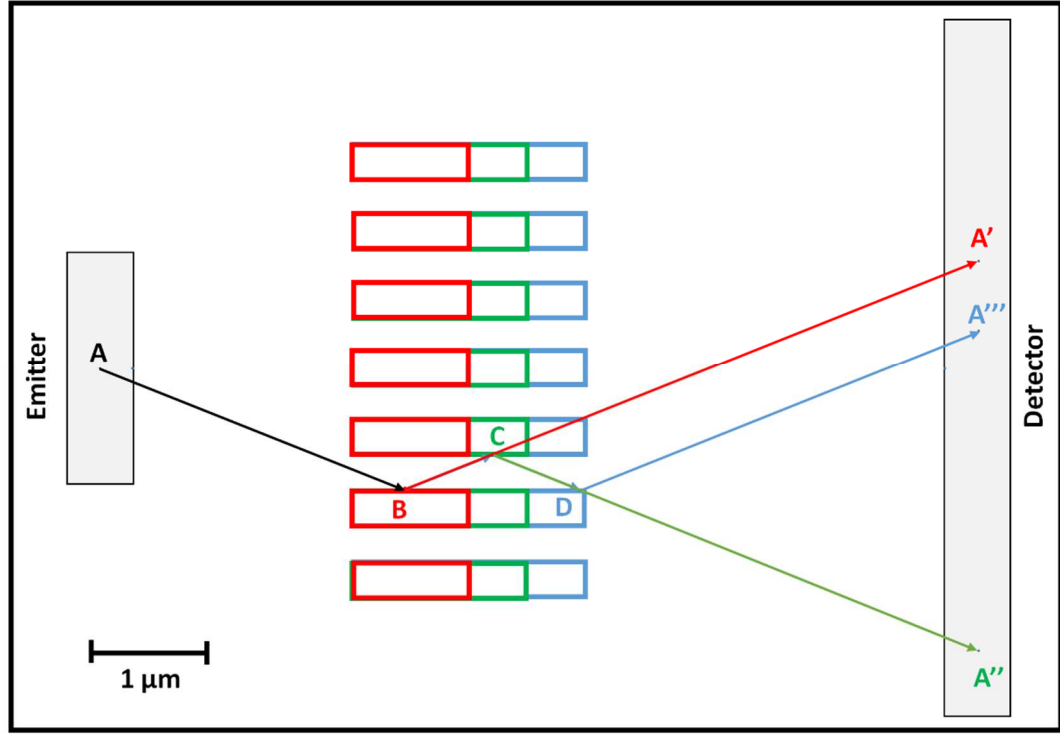
**Figure D.4: Measured signal produced by phonon transmission at  $V_s = 2.0$  mV through un-etched (bulk) Si mesa.** Signal values (vertical axis) are identical to those shown in **Figure 5.5** of paper: measured phonon transmission through mesa, presumed to be the sum of backscatter plus line-of-sight ballistic contributions,  $\dot{n}_{bs} + \dot{n}_{los}(s \rightarrow d)$ . Vertical error bars are standard error of signal average in each measurement. Horizontal axis is calculated transmission factor  $\Gamma_{los}(s \rightarrow d) \times 10^3$  for line-of-sight ballistic transmission through silicon mesa. Calculated values for closed

green circles were obtained using MC approach and calculated values for open purple diamonds were obtained using Analytic model. Both approaches include phonon focusing (PF) effects in single crystalline Si. Dashed green and purple lines correspond respectively to linear fits for closed green circles and open purple diamonds. Open red triangles: independent measurements of backscatter signal  $\dot{n}_{bs}$  on the same sample chips.



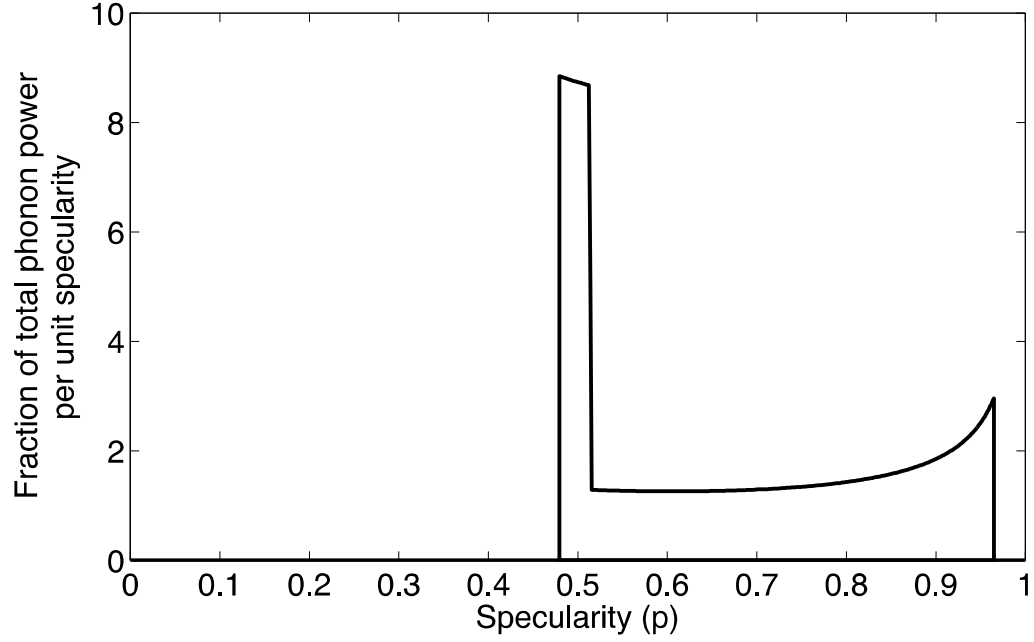
**Figure D.5: Measured signal produced by phonon transmission through ten different arrays of silicon nanosheets for phonon peak frequency of 400 GHz.**

Signal values (vertical axis) and error bars are identical to those shown in **Figure 5.6** of paper: measured phonon transmission through mesa, with estimated backscatter signal  $\dot{n}_{bs}$  subtracted. Circles, arrays A to C (widths  $\sim 0.15 \mu\text{m}$ ). Squares, arrays D to G (widths  $\sim 0.13 \mu\text{m}$ ). Triangles, arrays H to J (widths  $\sim 0.35 \mu\text{m}$ ) (see **Table 5.1** of the maintext). Horizontal axis line-of-sight ballistic transmission factors  $\Gamma_{los}(s \rightarrow d) + \Gamma_{los}(s \rightarrow ns \rightarrow d)$  are calculated using MC approach (closed green symbols) or Analytic model (open purple symbols). Both approaches include phonon focusing effects in single crystalline Si.

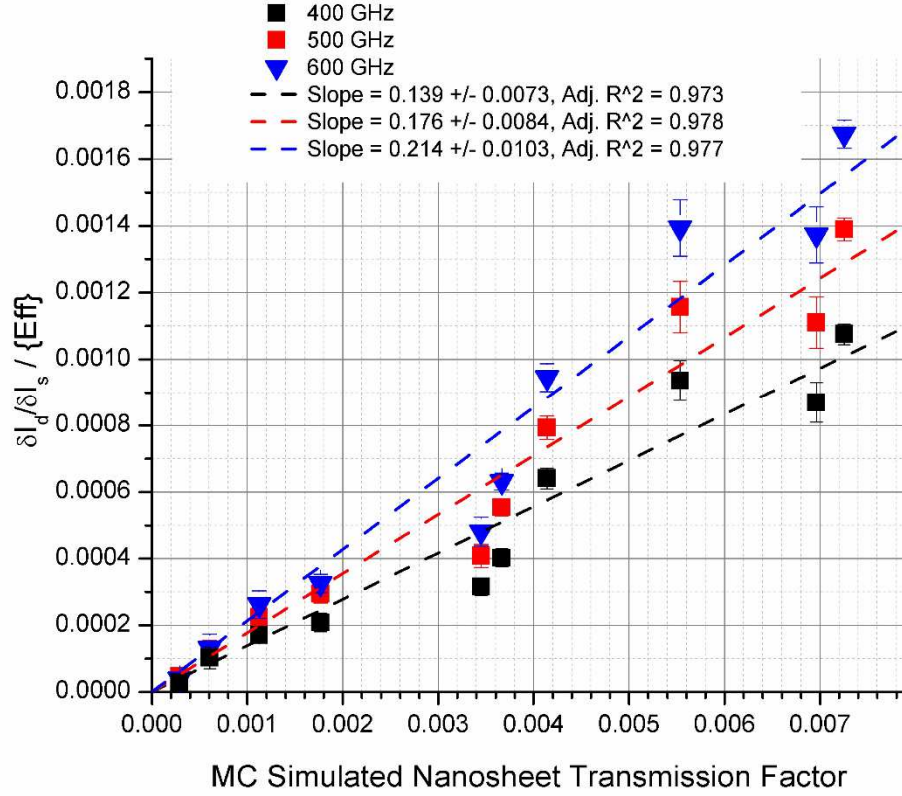


**Figure D.6: Test case performed to validate the MC calculation of phonon surface interaction.** Phonons are generated from a fixed point (A) source and the phonon vectors are also fixed to appropriate values to make all phonons hit point (B). Point (B) is located on the side wall of one of the 1 μm nanosheets (marked by red color). If the nanosheets are made to be 1.5 μm long (marked by green color) phonons specularly reflecting from point (B) hit point (C). If the nanosheets are made to be 2 μm long (marked by blue color) phonons specularly reflecting from point (C) hit point (D). Based on geometric calculations the phonons specularly reflecting from nanosheet sidewalls and hitting the detector are expected to land on points (A'), (A'') and (A''') for 1, 1.5 and 2 μm long nanosheets respectively. The number of phonons that land on these three points can be analytically calculated by  $m \times p^n$  where  $m$  is number of generated phonons,  $p$  is specularity of the nanosheet sidewalls and  $n$  is number of phonon surface interactions between emission and detection. In MC simulations, the number of

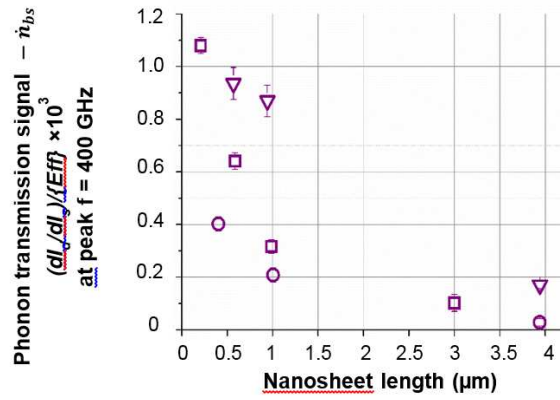
phonons hitting points (A'), (A'') and (A''') are counted and compared with analytical calculations. See **Table D.4** for this comparison and error analysis.



**Figure D.7: Distribution of specularity values predicted by Ziman expression for a 1 nm rough surface interacting with STJ emitted phonons at peak frequency of 400 GHz.**



**Figure D.8: Zero-specularity model of transmission through nanosheet arrays, plotted against transmission measurements for peak frequencies 400, 500, 600 GHz.** The horizontal axis values are phonon transmission factors  $[\Gamma_{los}(s \rightarrow ns \rightarrow d) + \Gamma_{sc}(s \rightarrow ns \rightarrow d) + \Gamma_{los}(s \rightarrow d)] \times 10^3$ , calculated using the MC method assuming specularity  $p = 0$  for phonon scattering at the nanosheet sidewalls, and are identical to those of the solid-symbol points in **Figure 5.6** of the main text. Vertical values are taken from the data in **Figure D.2** at the corresponding phonon spectral peak frequency. Estimated backscatter contribution  $\dot{n}_{bs}$  has been subtracted from all signal values. The 400 GHz datapoints in this plot are identical to the solid-symbol points in **Figure 5.6** of the main text. Linear fit parameters are indicated in the legend at top of the figure.



**Figure D.9: Measurements of phonon transmission through arrays of silicon nanosheets, using phonons of peak frequency 400 GHz.** Signal values and error bars are identical to those in **Figure 5.6** of main text. Horizontal axis is length of nanosheets in each array. Circles: arrays A to C. Squares: arrays D to G. Triangles: arrays H to J. (See **Table 5.1**, main text.).



Line Fit Range (160-300 GHz) / Line Fit Range( $\geq 300$ GHz)			
Transport (through)	Slope	Intercept	Adjusted- $R^2$
Bulk	3.71E-6/3.71E-6	-2.09E-4/-2.16E-4	0.9991/0.9994
0.2 $\mu\text{m}$ long Nanosheets (type D)	3.0E-6/2.88E-6	-1.26E-4/-1.01E-4	0.9976/0.9996
0.6 $\mu\text{m}$ long nanosheets (type E)	1.64E-6/1.58E-6	-2.63E-5/-2.29E-5	0.9917/0.9989
1 $\mu\text{m}$ long nanosheets (type F)	9.13E-7/7.50E-7	-1.81E-5/1.82E-5	0.9810/0.9918
3 $\mu\text{m}$ long nanosheets (type G)	2.47E-7/1.77E-7	2.50E-5/6.32E-5	0.5996/0.7570

**Table D.1: Results from linear fits of the phonon transmission signal.** The fitting was done for two different frequency ranges: Phonon signal between 160- 300 GHz, and phonon signal above 300 GHz. The fit lines appear in **Figure 5.3** of the main text. See **Table 5.1** of main text for nanosheet array dimensions.

$C_{11}$	$1.7750 \times 10^{11} \text{ (kg m}^{-1} \text{ s}^{-2}\text{)}$
$C_{12}$	$0.745 \times 10^{11} \text{ (kg m}^{-1} \text{ s}^{-2}\text{)}$
$C_{44}$	$0.807 \times 10^{11} \text{ (kg m}^{-1} \text{ s}^{-2}\text{)}$
Density	$2.33 \times 10^3 \text{ (kg m}^{-3}\text{)}$

**Table D.2: The values for elastic constants and density of Si that are used in the phonon focusing algorithm of the MC Approach.**

Measured Spectrometer #	Mesa Width ( $\mu\text{m}$ )	Source STJ Width ( $\mu\text{m}$ )	Source STJ Fraction on Mesa Sidewall	Det. Width ( $\mu\text{m}$ )	Mid-line of Emitter-to-Detector Line-of-sight. Angle Relative to Si [110] (Deg.)	Source STJ $R_N$ ( $\Omega$ )	Det. STJ $R_N$ ( $\Omega$ )	Calculated Ballistic Transmission Factor
1	6.84	0.90	0.32	3.13	3	2531	182	0.0050
2	6.81	0.97	0.34	2.97	24	2351	185	0.0029
3	6.84	0.80	0.31	2.85	48	2400	186	0.0018
4 <sup><math>\alpha</math></sup>	9.81	0.64	0.29	3.00	2	3915	197	0.0023
5 <sup><math>\beta</math></sup>	9.81	0.65	0.28	6.00	3	4057	202	0.0042
6	15.16	0.66	0.28	3.00	1	4139	200	0.0011
7	15.20	0.86	0.33	5.93	1	4056	207	0.0023
8	6.81	0.81	0.33	5.85	4	3590	207	0.0085
9 <sup><math>\beta</math></sup>	9.81	0.65	0.28	6.00	3	4051	204	0.0042
10 <sup><math>\alpha</math></sup>	9.81	0.64	0.29	3.00	2	3913	198	0.0023
11	9.88	0.65	0.31	6.00	7	4094	194	0.0045
12	9.84	0.54	0.25	5.89	7	3631	187	0.0036

**Table D.3: Geometric configurations of phonon sources and detectors in bulk**

**transmission measurements, with corresponding normal state resistance values.**

The phonon spectrometers marked with  $\alpha$  and  $\beta$  indicate repeated measurements of the same two spectrometers during two separate cooldowns of our cryostat.

Nanosheet Length	# Emitted Phonons	# Phonon Surface Interactions	Points Where Phonons are Expected to Land on the Detector	Specularity of Nanosheet Sidewalls	#Phonons that are expected to land on point A, A' and A''	# Phonons that land on point A, A' and A''' in MC Simulations	% Error for MC simulations
1 $\mu\text{m}$	1000000	1	A'	0.5	$0.5^1 * 1000000 = 500000$	499963	-0.0074
				1	$1^1 * 1000000 = 1000000$	1000000	0
1.5 $\mu\text{m}$	1000000	2	A''	0.5	$0.5^2 * 1000000 = 250000$	249511	-0.1956
				1	$1^2 * 1000000 = 1000000$	1000000	0
2 $\mu\text{m}$	1000000	3	A'''	0.5	$0.5^3 * 1000000 = 125000$	125491	0.3928
				1	$1^3 * 1000000 = 1000000$	1000000	0

**Table D.4: Number of phonons hitting points (A'), (A'') and (A''') in MC**

**simulations of 3 different test cases described in Figure D.6 and their comparison with analytical calculations (See captions of Figure D.6). Error values for MC simulations are reported in the last column.**

## D.6. Supplementary Methods: Monte Carlo C++ Code

```
// Phonon_Spectrometry_Simulation.cpp : Defines the entry point for the console application.
// Written by MAHMUT AKSIT - First version compiled on 4/11/2012 - Final version compiled on:
// 12/9/2013.

#include "stdafx.h"
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include "randgen.h"
#include <stdio.h>
#include "tvector.h"
#include <math.h>
#include <cmath>
#include <iomanip>
#include "strutils.h"
#include "prompt.h"
#include <limits.h>
#include <time.h>
#include "tmatrix.h"

#define PI 3.14159265
int bel80count = 0; // Belongs to advanced surface simulation algorithm
int totafmcount=0; // Belongs to advanced surface simulation algorithm
/* FULL SYSTEM MONTE CARLO SIMULATION OF PHONON SPECTROMETRY THROUGH NANOSHEETS

written by MAHMUT AKSIT for the phonon spectrometry project of Robinson Group-Cornell MSE*/

struct PLANE // Plane data structures definition
{
    int REFNUM,REGDETSYS,NSASSOC;
    /* REFNUM: Plane reference number, REGDETSYS: Regular plane, detector plane or system boundary
    plane,
    NSASSOC: Nanasheet associated with plane (planes are numbered starting from 1 and plane 1 is
    the center plane,
    plane 2 is to the left of it and plane 3 is to the right, even planes to the left, odd planes
    to the right
    */
    string PLADISC; /* String for describing the plane in words - generator plane, detector plane,
    nanosheet sidewall etc.*/
    double XLIMLOW, XLIMHIGH, YLIMLOW, YLIMHIGH, ZLIMLOW, ZLIMHIGH; /* X,Y,Z limits of the plane
    in 3D space*/
    double XPARAM, YPARAM, ZPARAM, PCONST; /* X,Y,Z parameters of the plane */
    double NIDIRC, NJDIRC, NKDIRC; /* unit i,j,k vector to define the normal vector of the plane*/
    double ROUGH; /* Roughness of the plane */
};

/* This function generates new planes as described by "strcut PLANE function". The inputs are
named the same way with the sub-variables for PLANE struct */
PLANE newplane(int refnum, string pladisc, double xlimlow, double xlimhigh, double
ylimlow, double ylimhigh, double zlimlow, double zlimhigh, double xparam, double yparam,
double zparam, double pconst, double nidirc, double njdirc, double nkdir, int regdetsys,
double rough, int nsassoc)
{
    PLANE plane;
    plane.REFNUM=refnum;
    plane.PLADISC=pladisc;
    plane.XLIMLOW=xlimlow;
    plane.XLIMHIGH=xlimhigh;
    plane.YLIMLOW=ylimlow;
    plane.YLIMHIGH=ylimhigh;
    plane.ZLIMLOW=zlimlow;
    plane.ZLIMHIGH=zlimhigh;
```

```

plane.XPARAM=xparam;
plane.YPARAM=yparam;
plane.ZPARAM=zparam;
plane.PCONST=pconst;
plane.NIDIRC=nidirc;
plane.NJDIRC=njdirc;
plane.NKDIRC=nkdirc;
plane.REGDETSYS=regdetsys; // 1 regular rough wall, 2 detector wall, 3 sys boundary, 4
counter, 5 useless
plane.ROUGH=rough;
plane.NSASSOC=nsassoc;//k for nanosheets, -1 for the gap walls, 0 for all others
return plane;
}

struct PHANGREP //struct for phonon nanosheet entrance exit counts and angles.
{
    int INCNSIND,INCNSTYPE; /* INCNSIND: the index number of the plane that the phonon interacts
with, INCNSTYPE: phonon nanosheet interaction type */
    double XPOS, YPOS, ZPOS, IVEC, JVEC, KVEC; /* Phonon-surface interaction positions x,y,z and
direction of the phonon in i, j ,k */
};

// Function for defining new PHANGREP struct
PHANGREP newphforangrep(int incnsind,int incnstype, double xpos, double ypos, double zpos,
double ivec, double jvec, double kvec)
{
    PHANGREP phangrep;
    phangrep.INCNSIND=incnsind;
    phangrep.INCNSTYPE=incnstype; //phonon entered ns=1, phonon exited ns=2, phonon entered ns
then hit the ns wall=3, phonon entered ns then hit the ns exit=4, phonon entered ns then hit
the det=5, phonon entered then hit ns bottom then the sys boundary=6, phonon entered ns than
hit the ns exit then det=7, phonon originated from ns bottom then exited ns then hit the
det=8, no history of entering ns but interacted with ns wall then exited=9, history of
entering ns and interaction with ns wall then exited=10, phonon that entered from open bottom
and directly hit ns exit=11, phonon with ns entrance history originated from ns wall then
exited from ns open bottom then hit sys boundary=12, phonon without ns entrance history
originated from ns wall then exited from ns open bottom then hit sys boundary=13, phonon
originated from ns wall then exited ns then hit detector with ns entrance history=14, phonon
originated from ns wall then exited ns then hit detector without ns entrance history=15,
phonon originated from ns bottom then hit ns wall=16, phonon originated from ns wall then hit
the ns bottom then detector with history of ns entrance=17, phonon originated from ns wall
then hit the ns bottom then detector without history of ns entrance=18
    phangrep.XPOS=xpos;
    phangrep.YPOS=ypos;
    phangrep.ZPOS=zpos;
    phangrep.IVEC=ivec;
    phangrep.JVEC=jvec;
    phangrep.KVEC=kvec;
    return phangrep;
}

double ferf(double x) // calculates very good approximate error function values-fast
{
    // constants
    double a1 = 0.254829592;
    double a2 = -0.284496736;
    double a3 = 1.421413741;
    double a4 = -1.453152027;
    double a5 = 1.061405429;
    double p = 0.3275911;

    // Save the sign of x
    int sign = 1;

```

```

    if (x < 0)
        sign = -1;
    x = fabs(x);

    // A&S formula 7.1.26
    double t = 1.0/(1.0 + p*x);
    double y = 1.0 - (((((a5*t + a4)*t) + a3)*t + a2)*t + a1)*t*exp(-x*x);

    return sign*y;
}

double erf(double t,int erfit) // calculates very good approximate error function values-slow
{
    RandGen randxy;
    double x,y,curv;
    int integcount=0;
    int i;
    for(i=0; i<=erfit; i++)
    {
        x= randxy.RandReal(0,t);
        y= randxy.RandReal(0,1);
        curv=exp((-1.0)*double(x)*double(x));
        if (curv != curv)
        {
            cout << "indefinite value for curv"<<endl;
        }

        if (y <= curv)
        {
            integcount=integcount+1;
        }
    }
    double erfvalue=t*1.12838*(double(integcount)/double(erfit));
    return erfvalue;
}

/*Function for definition of uniform phonon frequency distribution*/
void unifreqdist(tvector<double> &freqlist, tvector<int> &tempphononsperf_vec, int phononstot,
double freqlow, double freqhigh, int freqnum)
{
    int k;
    int phononsperf;
    double freqstepsize;
    phononsperf=double(phononstot)/double(freqnum);
    freqstepsize=(freqhigh-freqlow)/double(freqnum-1);

    for (k=0; k<freqnum;k++)
    {
        freqlist.push_back(freqlow+double(freqstepsize*double(k)));
        tempphononsperf_vec.push_back(phononsperf);
    }
}

/* Function that invert matrices.*/
tmatrix<double> invertmat(const tmatrix<double> & inmatrix)
{
    int i,j;
    double determinant=0;
    tmatrix<double> invmatrix(3,3);
    // Activate codes in the next three lines to see the matrix
    /*cout << "matrix" << inmatrix[0][0] << " " <<inmatrix[0][1] << " " <<inmatrix[0][2] <<
endl;
    cout << "matrix" << inmatrix[1][0] << " " <<inmatrix[1][1] << " " <<inmatrix[1][2] << endl;
    cout << "matrix" << inmatrix[2][0] << " " <<inmatrix[2][1] << " " <<inmatrix[2][2] << endl;

```

```

*/
for(i=0;i<3;i++)
{
    determinant = determinant + (inmatrix[0][i]*(inmatrix[1][(i+1)%3]*inmatrix[2][(i+2)%3] -
inmatrix[1][(i+2)%3]*inmatrix[2][(i+1)%3]));
}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        invmatrix[i][j] = ((inmatrix[(i+1)%3][(j+1)%3] * inmatrix[(i+2)%3][(j+2)%3]) -
(inmatrix[(i+1)%3][(j+2)%3]*inmatrix[(i+2)%3][(j+1)%3]))/determinant;
    }
}

// Activate codes in the next three lines to see the matrix
/*cout << "inverse matrix" << invmatrix[0][0] << " " << invmatrix[0][1] << " "
<< invmatrix[0][2] << endl;
cout << "inverse matrix" << invmatrix[1][0] << " " << invmatrix[1][1] << " "
<< invmatrix[1][2] << endl;
cout << "inverse matrix" << invmatrix[2][0] << " " << invmatrix[2][1] << " "
<< invmatrix[2][2] << endl;
*/
return invmatrix;
}

/*Function for solving cubic equations. Useful for PFF calculations. Each root is for
different phonon polarizations*/
void cubeqsolv(double &kx, int polar, double a, double b, double c, double d)
{
    double datemp,e,f,g,h,i,j,k,l,m,n,p,r,rtemp,s,t,ttemp,u,x1,x2,x3,x23i,kx1, kx2, kx3;
    int w,negativ;
    x23i=0;
    e=2.7182818284590;
    f=((double(3)*c/a)-(b*b/(a*a)))/double(3);
    g=((double(2)*b*b/(a*a*a))-(double(9)*b*c/(a*a))+double(27)*d/a)/double(27);
    h=(g*g/double(4))+(f*f*f/double(27));
    i=sqrt(((g*g/double(4))-h));
    j=exp(log10(i)/log10(e)/double(3));
    k=acos((double(-1))*(g/(double(2)*i)));
    l=j*(double(-1));
    m=cos(k/double(3));
    n=sqrt(double(3))*sin(k/double(3));
    p=(b/(double(3)*a))*(double(-1));
    r=((double(-1))*(g/double(2)))+sqrt(double(h));
    if (r<0)
    {
        rtemp=-r;
        s=-pow(rtemp,(double(1)/double(3)));
    }else
    {
        rtemp=r;
        s=pow(rtemp,(double(1)/double(3)));
    }
    t=(double(-1))*(g/double(2))-sqrt(double(h));

    if (t<0)
    {
        ttemp=-t;
        u=-pow(ttemp,(double(1)/double(3)));
    }else
    {
        ttemp=t;
        u=pow(ttemp,(double(1)/double(3)));
    }
}

```

```

}

if (h>0) w=1;
if (h<=0) w=3;
if ((f==0) && (g==0) && (h==0)) w=2;

switch (w){
case 1:
x1=(s+u)-(b/(double(3)*a));
x2=(double(-1))*(s+u)/double(2)-(b/(double(3)*a));
x3=x2;
x23i=(s-u)*sqrt(double(3))/double(2);
break;
case 2:
if ((d/a)<0)
{
datemp=-(d/a);
x1=-pow(datemp,(double(1)/double(3)))*(double(-1));
}else
{
datemp=(d/a);
x1=pow(datemp,(double(1)/double(3)))*(double(-1));
}
x2=x1;
x3=x1;
break;
case 3:
x3=double(2)*j*cos(k/double(3))-(b/(double(3)*a));
x1=1*(m+n)+p;
x2=1*(m-n)+p;
break;
}
// Activate codes in the next four lines to see the complex roots
/*cout<<x1<<endl;
cout<<x2<<endl;
cout<<x3<<endl;
cout<<x23i<<endl;*/

kx1=sqrt(x1);
kx2=(1.0/sqrt(double(2)))*sqrt(sqrt((x2*x2)+(x23i*x23i))+x2);
kx3=(1.0/sqrt(double(2)))*sqrt(sqrt((x3*x3)+(x23i*x23i))+x3);

// Activate codes in the next four lines to see the roots
/*cout<<kx1<<endl;
cout<<kx2<<endl;
cout<<kx3<<endl;*/

// Determine the phonon polarization based on pre-assigned polar value
if (polar==1) // L polarization
{
kx=kx1;
}else if (polar==2) // FT polarization
{
kx=kx2;
}
else if (polar==3) // ST polarization
{
kx=kx3;
}

}

/* Function that transforms the k vector into group velocity vector based on PFF
calculations*/

```



```

void pff(double &prekxex, double &prekyex, double &prekzex, double freq,int ppolar)
{
    double lambda, angfreq, prekvecmag, m,n, kx, ky, kz,prekx,preky,prekz, kvecmag, Q, S, R, T, U,
    V, Qc, Sc, Rc, Tc, Uc, Vc, Qx, Sx, Rx, Tx, Ux, Vx, Qy, Sy, Ry, Ty, Uy, Vy, Qz, Sz, Rz, Tz, Uz,
    Vz, ro, nuc, nud, nue, e, d, c, ex, dx, cx, ey, dy, cy, ez, dz, cz, N, C11, C12, C44, Vgx,
    Vgy, Vgz, Vgvecmag, phIvec, phJvec, PhKvec;

    //vector transform into pff coordinates for crystal orientation 110 - In order to activate,
    convert to active text and convert the other to passive text
    prekx=((1.0/sqrt(double(2.0)))*prekxex)+((1.0/sqrt(double(2.0)))*prekyex);
    preky=(((-1.0)/sqrt(double(2.0)))*prekxex)+((1.0/sqrt(double(2.0)))*prekyex);
    prekz=prekzex;
    //vector transform into pff coordinates for crystal orientation 110*/

    /*vector transform into pff coordinates for crystal orientation 100 - In order to activate,
    convert to active text and convert the other to passive text
    prekx=prekxex;
    preky=prekyex;
    prekz=prekzex;
    //vector transform into pff coordinates for crystal orientation 100*/

    /*vector transform into pff coordinates with theta amount rotation around l,m,n vector - In
    order to activate, convert to active text and convert the other to passive text

    double prekxt,prekyt,prekzt;

    double lt, lnorm;
    double mt, mnorm;
    double nt, nnorm;
    double thero;

    lt=0;
    mt=0;
    nt=1;

    thero=45;

    lnorm=lt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
    mnorm=mt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
    nnorm=nt/sqrt((lt*lt)+(mt*mt)+(nt*nt));

    prekxt=((lnorm*lnorm*(1-
    cos(thero*(PI/180))))+cos(thero*(PI/180))*prekxex)+((mnorm*lnorm*(1-cos(thero*(PI/180))))-
    (nnorm*sin(thero*(PI/180)))*prekyex)+((lnorm*lnorm*(1-
    cos(thero*(PI/180))))+(mnorm*sin(thero*(PI/180)))*prekzex);
    prekyt=((lnorm*mnorm*(1-
    cos(thero*(PI/180))))+(nnorm*sin(thero*(PI/180)))*prekxex)+((mnorm*mnorm*(1-
    cos(thero*(PI/180))))+(cos(thero*(PI/180)))*prekyex)+((nnorm*mnorm*(1-cos(thero*(PI/180))))-
    (lnorm*sin(thero*(PI/180)))*prekzex);
    prekzt=((lnorm*nnorm*(1-cos(thero*(PI/180))))-
    (mnorm*sin(thero*(PI/180)))*prekxex)+((mnorm*nnorm*(1-
    cos(thero*(PI/180))))+(lnorm*sin(thero*(PI/180)))*prekyex)+((nnorm*nnorm*(1-
    cos(thero*(PI/180))))+(cos(thero*(PI/180)))*prekzex);

    lt=1;
    mt=-1;
    nt=0;

    thero=0;

    lnorm=lt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
    mnorm=mt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
    nnorm=nt/sqrt((lt*lt)+(mt*mt)+(nt*nt));

```

```

    prekx=((lnorm*lnorm*(1-cos(thero*(PI/180))))+cos(thero*(PI/180))*prekxt)+((mnorm*lnorm*(1-
cos(thero*(PI/180))))-(nnorm*sin(thero*(PI/180)))*prekyt)+(((nnorm*lnorm*(1-
cos(thero*(PI/180))))+(mnorm*sin(thero*(PI/180)))*prekzt);
    preky=((lnorm*mnorm*(1-
cos(thero*(PI/180))))+(nnorm*sin(thero*(PI/180)))*prekxt)+((mnorm*mnorm*(1-
cos(thero*(PI/180))))+cos(thero*(PI/180))*prekyt)+(((nnorm*mnorm*(1-cos(thero*(PI/180))))-
(lnorm*sin(thero*(PI/180)))*prekzt);
    prekz=((lnorm*nnorm*(1-cos(thero*(PI/180))))-
(mnorm*sin(thero*(PI/180)))*prekxt)+((mnorm*nnorm*(1-
cos(thero*(PI/180))))+(lnorm*sin(thero*(PI/180)))*prekyt)+(((nnorm*nnorm*(1-
cos(thero*(PI/180))))+(cos(thero*(PI/180)))*prekzt);

//vector transform into pff coordinates with theta amount rotation around l,m,n vector*/

// Si elastic constants are defined for use in Christoffel equations
C11 = 1.7750*pow(double(10), double(11)); //(kg m-1 s-2)
C12 = 0.745*pow(double(10), double(11)); // (kg m-1 s-2)
C44 = 0.807*pow(double(10), double(11)); //(kg m-1 s-2)

ro=2.33*pow(double(10), double(3)); // kg m-3

// phonon angular frequency calculated
angfreq=double(2.0)*double(PI)*freq;

// lambda for Christoffel equations-see Jakata thesis
lambda=ro*angfreq*angfreq;

if (prekx==0)
{
    prekx=prekx+0.000001;
}

/*prekvecmag=sqrt((prekx*prekx)+(preky*preky)+(prekz*prekz));

prekx=prekx/prekvecmag;
preky=preky/prekvecmag;
prekz=prekz/prekvecmag;*/

m=preky/prekx;
n=prekz/prekx;

//variables for Christoffel equations see Jakara thesis
Qc=C11+(C44*((m*m)+(n*n)));
Sc=(C11*n*n)+(C44*(1+(m*m)));
Rc=(C11*m*m)+(C44*(1+(n*n)));
Tc=m*(C12+C44);
Uc=n*(C12+C44);
Vc=m*n*(C12+C44);

nuc=(Qc+Rc+Sc);
nud=(Tc*Tc)+(Uc*Uc)+(Vc*Vc)-(Sc*Qc)-(Sc*Rc)-(Qc*Rc);
nue=(Qc*Rc*Sc)-(Vc*Vc*Qc)-(Tc*Tc*Sc)-(Uc*Uc*Rc)+(2*Tc*Uc*Vc);

cubeqsolv(kx,ppolar, nue, nud*lambda, nuc*lambda*lambda, double(-1)*lambda*lambda*lambda);

// Group velocity vector is inversed if the phonon points toward the wall
if (prekx<0)
{
    kx=-kx;
}

ky=kx*m;
kz=kx*n;

```

```

kvecmag=sqrt((kx*kx)+(ky*ky)+(kz*kz));
//cout<<"kvec magnitude: " << kvecmag << endl;

// math performed below to solve Christoffel equations
Qx=double(2.0)*C11*kx;
Qy=double(2.0)*C44*ky;
Qz=double(2.0)*C44*kz;

Rx=double(2.0)*C44*kx;
Ry=double(2.0)*C11*ky;
Rz=double(2.0)*C44*kz;

Sx=double(2.0)*C44*kx;
Sy=double(2.0)*C44*ky;
Sz=double(2.0)*C11*kz;

Tx=ky*(C12+C44);
Ty=kx*(C12+C44);
Tz=0;

Ux=kz*(C12+C44);
Uy=0;
Uz=kx*(C12+C44);

Vx=0;
Vy=kz*(C12+C44);
Vz=ky*(C12+C44);

Q=Qc*pow(kx, double(2));
R=Rc*pow(kx, double(2));
S=Sc*pow(kx, double(2));

T=Tc*pow(kx, double(2));
U=Uc*pow(kx, double(2));
V=Vc*pow(kx, double(2));

e=nue*pow(kx, double(6));
d=nud*pow(kx, double(4));
c=nuc*pow(kx, double(2));

N=((double(3.0)*lambda*lambda)-(double(2.0)*lambda*c)-d)*double(2)*ro*angfreq;

ex=(Qx*R*S)+(Q*((Rx*S)+(R*Sx)))-(((double(2)*V*Vx*Q)+(V*V*Qx))-((double(2)*T*Tx*S)+(T*T*Sx))-
((double(2)*U*Ux*R)+(U*U*Rx)))+(double(2)*(Tx*U*V+(T*((Ux*V)+(U*Vx)))));
dx=(double(2)*T*Tx)+(double(2)*U*Ux)+(double(2)*V*Vx)-((Sx*Q)+(S*Qx))-((Sx*R)+(S*Rx))-
((Qx*R)+(Q*Rx));
cx=Qx+Rx+Sx;

ey=(Qy*R*S)+(Q*((Ry*S)+(R*Sy)))-(((double(2)*V*Vy*Q)+(V*V*Qy))-((double(2)*T*Ty*S)+(T*T*Sy))-
((double(2)*U*Uy*R)+(U*U*Ry)))+(double(2)*(Ty*U*V+(T*((Uy*V)+(U*Vy)))));
dy=(double(2)*T*Ty)+(double(2)*U*Uy)+(double(2)*V*Vy)-((Sy*Q)+(S*Qy))-((Sy*R)+(S*Ry))-
((Qy*R)+(Q*Ry));
cy=Qy+Ry+Sy;

ez=(Qz*R*S)+(Q*((Rz*S)+(R*Sz)))-(((double(2)*V*Vz*Q)+(V*V*Qz))-((double(2)*T*Tz*S)+(T*T*Sz))-
((double(2)*U*Uz*R)+(U*U*Rz)))+(double(2)*(Tz*U*V+(T*((Uz*V)+(U*Vz)))));
dz=(double(2)*T*Tz)+(double(2)*U*Uz)+(double(2)*V*Vz)-((Sz*Q)+(S*Qz))-((Sz*R)+(S*Rz))-
((Qz*R)+(Q*Rz));
cz=Qz+Rz+Sz;

// Group velocity vector is defined
Vgx=(ex+(dx*lambda)+(cx*lambda*lambda))/N;
Vgy=(ey+(dy*lambda)+(cy*lambda*lambda))/N;

```

```

Vgz=(ez+(dz*lambda)+(cz*lambda*lambda))/N;
Vgvecmag=sqrt((Vgx*Vgx)+(Vgy*Vgy)+(Vgz*Vgz));

//cout << "Group velocities for x: " << Vgx << " for y: " << Vgy << " for z: " << Vgz << ".
Magnitude of group velocity vector: "<< Vgvecmag << endl;

// unit vector for phonon direction is defined: - In order to activate, convert to active text
and convert the other to passive text

prekx=Vgx/Vgvecmag;
preky=Vgy/Vgvecmag;
prekz=Vgz/Vgvecmag;

//vector transform back into phonon spectrometry coordinates for crystal orientation 110
prekxex=((1.0/sqrt(double(2.0)))*prekx)+((-1.0)/sqrt(double(2.0))*preky);
prekyex=((1.0/sqrt(double(2.0)))*prekx)+(1.0/sqrt(double(2.0))*preky);
prekzex=prekz;
//vector transform back into phonon spectrometry coordinates for crystal orientation 110*/

/*vector transform back into phonon spectrometry coordinates for crystal orientation 100
prekxex=prekx;
prekyex=preky;
prekzex=prekz;
//vector transform back into phonon spectrometry coordinates for crystal orientation 100*/

/*vector transform back into phonon spectrometry coordinates with theta amount rotation around
l,m,n vector
lt=1;
mt=-1;
nt=0;

thero=0;

lnorm=lt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
mnorm=mt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
nnorm=nt/sqrt((lt*lt)+(mt*mt)+(nt*nt));

prekxt=((lt*lt*(1-cos(thero*(PI/180)))+cos(thero*(PI/180))*prekx)+((lt*mt*(1-
cos(thero*(PI/180)))+(nt*sin(thero*(PI/180))*preky)+((lt*nt*(1-cos(thero*(PI/180)))-
(mt*sin(thero*(PI/180))*prekz);
prekyt=((mt*lt*(1-cos(thero*(PI/180)))-(nt*sin(thero*(PI/180))*prekx)+((mt*mt*(1-
cos(thero*(PI/180)))+(cos(thero*(PI/180))*preky)+((mt*nt*(1-
cos(thero*(PI/180)))+(lt*sin(thero*(PI/180))*prekz);
prekzt=((nt*lt*(1-cos(thero*(PI/180)))+(mt*sin(thero*(PI/180))*prekx)+((nt*mt*(1-
cos(thero*(PI/180)))-(lt*sin(thero*(PI/180))*preky)+((nt*nt*(1-
cos(thero*(PI/180)))+(cos(thero*(PI/180))*prekz);

lt=0;
mt=0;
nt=1;

thero=-45.0;

lnorm=lt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
mnorm=mt/sqrt((lt*lt)+(mt*mt)+(nt*nt));
nnorm=nt/sqrt((lt*lt)+(mt*mt)+(nt*nt));

prekxex=((lt*lt*(1-cos(thero*(PI/180)))+cos(thero*(PI/180))*prekxt)+((lt*mt*(1-
cos(thero*(PI/180)))+(nt*sin(thero*(PI/180))*prekyt)+((lt*nt*(1-cos(thero*(PI/180)))-
(mt*sin(thero*(PI/180))*prekzt);
prekyex=((mt*lt*(1-cos(thero*(PI/180)))-(nt*sin(thero*(PI/180))*prekxt)+((mt*mt*(1-
cos(thero*(PI/180)))+(cos(thero*(PI/180))*prekyt)+((mt*nt*(1-
cos(thero*(PI/180)))+(lt*sin(thero*(PI/180))*prekzt);

```

```

    prekzex=((nt*lt*(1-cos(thero*(PI/180)))+(mt*sin(thero*(PI/180)))*prekxt)+((nt*mt*(1-
cos(thero*(PI/180)))-(lt*sin(thero*(PI/180)))*prekyt)+((nt*nt*(1-
cos(thero*(PI/180)))+(cos(thero*(PI/180)))*prekzt);

    //vector transform back into phonon spectrometry coordinates with theta amount rotation around
    l,m,n vector*/

    //system ("pause");

}
// emitted phonon frequency distribution defined based on STJ phonon emission assumption.
Uniform until peak frequency. Ratio of peak phonons and tail phonons should be provided
void emsfreqdist(tvector<double> &freqlist, tvector<int> &tempphononsperf_vec, int phononstot,
double freqlow, double peakfreqems, double peakratio, int freqnum)
{
    int k;
    double freqstepsize;

    freqstepsize=(peakfreqems-freqlow)/double(freqnum-1); // define frequency step size

    double peakphonons=double(phononstot)*(peakratio);
    double tailphonons=double(phononstot)*(1.000-peakratio);

    for (k=0; k<(freqnum-1);k++)
    {
        freqlist.push_back(freqlow+double(freqstepsize*double(k))); // define frequency list based on
        emission distribution. Phonon 1 has this frequency, Phonon 2 has some other frequency etc.
        tempphononsperf_vec.push_back(int(tailphonons/double(freqnum-1)));
    }

    freqlist.push_back(peakfreqems);
    tempphononsperf_vec.push_back(int(peakphonons));

}
// Function for reporting phonon free path lengths, writes into a fpath vector
tvector<int> reportfpath(const tvector<double> &fpath, int phononsperf, double binsize, double
frequency, double &minfpath, int &numbins)
{
    int m,l;
    double maxfpath=0;

    if (fpath.size()>(phononsperf+1))
    {
        cout<<"Warning!!! There is a problem with the size of the freepath vector. It is larger than
number of phonons"<<endl;
        cout<<"Free Path Vector Size: " <<fpath.size()<<endl;
        cout<<"Number of phonons: " <<phononsperf<<endl;
        //system ("pause");
    }

    for(m=0; m<fpath.size(); m++)
    {
        if (fpath[m]>maxfpath)
        {
            maxfpath=fpath[m];
        }
    }
    minfpath=maxfpath;
    for (m=0; m<fpath.size(); m++)
    {
        if (fpath[m] < minfpath)
        {
            minfpath=fpath[m];
        }
    }
}

```

```

numbins=int((maxfpath-minfpath)/binsize)+1;
tvector<int> fpbin(numbins,0);

for (m=0; m<fpath.size(); m++)
{
    for (l=0;l<numbins;l++)
    {
        if ((fpath[m] > ((l*binsize)+minfpath)) && (fpath[m] < (((l+1)*binsize)+minfpath)))
        {
            fpbin[l]=fpbin[l]+1;
        }
    }
}

return fpbin;

}
// Prints list of plane normal vector magnitudes on screen
void listplvecmag(tvector<PLANE> &planelist)
{
    int plsize=planelist.size();
    int j;
    for (j=0; j < plsize; j++)
    {
        cout << j << "\t" << planelist[j].REFNUM << "\t" << "vector magnitude= " <<
sqrt(double((planelist[j].NIDIRC*planelist[j].NIDIRC)+(planelist[j].NJDIRC*planelist[j].NJDIRC
)+(planelist[j].NKDIRC*planelist[j].NKDIRC))) << endl;
    }
}

// outputs phonon direction vector magnitude - should always be 1.
void phvecmag(tvector<double> &phonon)
{
    cout << "phonon vector magnitude= " <<
sqrt(double((phonon[5]*phonon[5])+(phonon[6]*phonon[6])+(phonon[7]*phonon[7]))) << endl;
}
// outputs any vector magnitude
void vecmag(tvector<double> &anyvec)
{
    cout << "any vector magnitude= " <<
sqrt(double((anyvec[0]*anyvec[0])+(anyvec[1]*anyvec[1])+(anyvec[2]*anyvec[2]))) << endl;
}

//
//generates random vectors, checkpoints are inserted to observe indefinite number issues
void genrandvec(tvector<double> &randvec, double costhelim, double philim, int ifcos)
{
    RandGen randomv;
    double costheta= randomv.RandReal(costhelim,1);
    double theta=acos(costheta);
    /*double theta=(randomv.RandReal(0.0001,thetalim));*/
    double phi=(randomv.RandReal(0.000001,philim));

    if (theta!=theta)
    {
        /*cout << "Indefinite number issue in genrandvec algorithm. theta is indefinite"<< endl;
        system ("pause");*/
    }
    if (phi!=phi)
    {
        /*cout << "Indefinite number issue in genrandvec algorithm. phi is indefinite"<< endl;

```

```

    system ("pause");*/
}
double randnum=1;
if (ifcos==1) // Activates when Lambertian distribution is selected for phonon emission
{
    if (costhelim >= 0)
    {

        double a=cos(theta);
        int cntr=0;
        double cos2theta= randomv.RandReal(-1,1);
        theta=acos(cos2theta)/2.0;

    }
    else
    {
        cout <<"cos(theta) distribution is not valid for theta limit higher than 90" << endl;
    }
}

randvec.push_back(double(sin(theta)*cos(phi*PI/180)));
randvec.push_back(double(sin(theta)*sin(phi*PI/180)));
randvec.push_back(double(cos(theta)));

}

//generates random vectors on a plane using genrandvec function. the vector can not point
towards the plane i.e. the angle between plane normal vector and phonon can not be larger than
90
void genrandveconpl(tvector<double> &randveconpl, const PLANE genplane)
{
    tvector<double> randvec(0);
    genrandvec(randvec, -1,360,0);
    double rveconpli = (genplane.NJDIRC*randvec[2])-(genplane.NKDIRC*randvec[1]);
    double rveconplj =(genplane.NKDIRC*randvec[0])-(genplane.NIDIRC*randvec[2]);
    double rveconplk =(genplane.NIDIRC*randvec[1])-(genplane.NJDIRC*randvec[0]);
    double rvecmag =
    sqrt(double((rveconpli*rveconpli)+(rveconplj*rveconplj)+(rveconplk*rveconplk)));

    // Below if conditions are to detect indefinite number incidents
    if (rveconpli!=rveconpli)
    {
        /*cout << "Indefinite number issue in diffscatt algorithm. rveconpli is indefinite"<< endl;
        system ("pause");*/
    }
    if (rveconplj!=rveconplj)
    {
        /*cout << "Indefinite number issue in genrandveconpl algorithm. rveconplj is indefinite"<<
        endl;
        system ("pause");*/
    }
    if (rveconplk!=rveconplk)
    {
        /*cout << "Indefinite number issue in genrandveconpl algorithm. rveconplk is indefinite"<<
        endl;
        system ("pause");*/
    }
    if (rvecmag!=rvecmag)
    {
        /*cout << "Indefinite number issue in genrandveconpl algorithm. rvecmag is indefinite"<<
        endl;
        system ("pause");*/
    }
}

```

```

randveconpl.push_back(rveconpli/rvecmag);
randveconpl.push_back(rveconplj/rvecmag);
randveconpl.push_back(rveconplk/rvecmag);
}
// This function is not actively used version of generatepls
void generatedefpls(tvector<double> &phonon, const tvector<PLANE> &planelist, double
frequency, double phrefnum, double phIvec, double phJvec, double phKvec)
{
    phonon[0]=(frequency);
    phonon[1]=(phrefnum);
    RandGen randoms;
    tvector<double> randveconpl(0);
    tvector<double> secveconpl(0);
    tvector<double> temprandvec(0);
    double ptxpos,ptypos, ptzpos;
    ptxpos = (planelist[4].XLIMHIGH+planelist[4].XLIMLOW)/2;
    ptypos = (planelist[4].YLIMHIGH+planelist[4].YLIMLOW)/2;
    if (planelist[4].ZPARAM==0)
    {
        ptzpos = (planelist[4].ZLIMHIGH+planelist[4].ZLIMLOW)/2;
    }
    else
    {
        ptzpos = (planelist[4].PCONST-(planelist[4].XPARAM*ptxpos)-
(planelist[4].YPARAM*ptypos))/planelist[4].ZPARAM;
    }

    phonon[2]=(ptxpos);
    phonon[3]=(ptypos);
    phonon[4]=(ptzpos);

    phonon[5]=(phIvec)/ sqrt(double((phIvec*phIvec)+(phJvec*phJvec)+(phKvec*phKvec)));
    phonon[6]=(phJvec)/ sqrt(double((phIvec*phIvec)+(phJvec*phJvec)+(phKvec*phKvec)));
    phonon[7]=(phKvec)/ sqrt(double((phIvec*phIvec)+(phJvec*phJvec)+(phKvec*phKvec)));
}

// writes defined planes in a vector called "planelist"
void defineplane(tvector<PLANE> &planelist, int refnum, string pladisc, double xlimlow,
double xlimhigh, double ylimlow, double ylimhigh, double zlimlow, double zlimhigh, double
xparam, double yparam, double zparam, double pconst, double nidirc, double njdirc, double
nkdir, int regdetsys, double rough, int nsassoc)
{
    planelist.push_back(newplane(refnum, pladisc,xlimlow, xlimhigh, ylimlow, ylimhigh, zlimlow,
zlimhigh, xparam, yparam, zparam, pconst, nidirc, njdirc, nkdir,regdetsys,rough, nsassoc));
}

// Regression plane normal vector calculated for afm defines surfaces. This is part of
advanced surface simulations trials. This is not actively used in MC simulations
tvector<double> regplnormvec(const tmatrix<double> &afmroughness, int rmatindx, int rmatindy,
int rrange)
{
    tmatrix<double> amatrix(3,3);
    tmatrix<double> invamatrix(3,3);
    tmatrix<double> bmatrix(3,1);
    tvector<double> normvec(3);
    double sumxx, sumxy, sumx, sumyy, sumy, sumxz, sumyz, sumz, n, ivec, jvec, kvec;
    int i, j;
    sumxx=0;
    sumxy=0;
    sumx=0;
    sumyy=0;
    sumy=0;
    sumz=0;
    sumxz=0;
    sumyz=0;

```



```

n=double(((2*rrange)+1)*((2*rrange)+1));

for (i=-rrange; i <= rrange; i++)
{
    for (j=-rrange; j <= rrange; j++)
    {
        sumxx= sumxx + double(rmatindx+i)*double(rmatindx+i);
        sumyy= sumyy + double(rmatindy+j)*double(rmatindy+j);
        sumxy= sumxy + double(rmatindx+i)*double(rmatindy+j);
        sumx= sumx + double(rmatindx+i);
        sumy= sumy + double(rmatindy+j);

        sumxz= sumxz +
double(rmatindx+i)*afmrroughness[rmatindx+i][rmatindy+j]*double(pow(double(10),double(9)));
        sumyz= sumyz +
double(rmatindy+j)*afmrroughness[rmatindx+i][rmatindy+j]*double(pow(double(10),double(9)));
        sumz= sumz + afmrroughness[rmatindx+i][rmatindy+j]*double(pow(double(10),double(9)));
    }
}

amatrrix[0][0]=sumxx;

amatrrix[0][1]=sumxy;
amatrrix[0][2]=sumx;
amatrrix[1][0]=sumxy;
amatrrix[2][0]=sumx;

amatrrix[1][1] = sumyy;
amatrrix[1][2] = sumy;
amatrrix[2][1] = sumy;

amatrrix[2][2] = n;

bmatrrix[0][0] = sumxz;
bmatrrix[1][0] = sumyz;
bmatrrix[2][0] = sumz;

invamatrix = invertmat(amatrrix);

ivec=0;
jvec=0;
kvec=0;

for(j=0;j<3;j++)
{
    ivec=ivec+invamatrix[0][j]*bmatrrix[j][0];
    jvec=jvec+invamatrix[1][j]*bmatrrix[j][0];
    kvec=kvec+invamatrix[2][j]*bmatrrix[j][0];
}

if (kvec < 0 )
{
    ivec=-ivec;
    jvec=-jvec;
    kvec=-kvec;
}

```

```

normvec[0]=ivec/double(sqrt(double((ivec*ivec)+(jvec*jvec)+(kvec*kvec))));
normvec[1]=jvec/double(sqrt(double((ivec*ivec)+(jvec*jvec)+(kvec*kvec))));
normvec[2]=kvec/double(sqrt(double((ivec*ivec)+(jvec*jvec)+(kvec*kvec))));
return normvec;
}

// Advanced surface simulation algorithm trial. It tries to simulation phonon-surface
interactions based AFM measure surface. This is not actively used in nanosheet phonon
transport MC simulations.
void surfsimafm(tvector<double> &phonon, PLANE crntplane, const tmatrix<double>
&afmroughness, int rrange, int ifpffafm)
{
double surfphx, surfphy, surfphz,surfphxorg, surfphyorg, surfphzorg, dotproln;
tvector<double> nafm(3);
int rmatindx, rmatindy, ppolar;
int reenter, n;
RandGen randoms;
if((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==1)&&(crntplane.NKDIRC==0))
{
//vector transform into surf coordinates of n-010 plane
surfphx=phonon[5];
surfphy=phonon[7]*double(-1);
surfphz=phonon[6];
//vector transform into surf coordinates of n-010 plane
}
else if ((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==--1)&&(crntplane.NKDIRC==0))
{
//vector transform into surf coordinates of n-0-10 plane
surfphx=phonon[5];
surfphy=phonon[7];
surfphz=phonon[6]*double(-1);
//vector transform into surf coordinates of n-0-10 plane
}
else if ((crntplane.NIDIRC==1)&&(crntplane.NJDIRC==0)&&(crntplane.NKDIRC==0))
{
//vector transform into surf coordinates of n-100 plane
surfphx=phonon[7]*double(-1);
surfphy=phonon[6];
surfphz=phonon[5];
//vector transform into surf coordinates of n-100 plane
} else if ((crntplane.NIDIRC==--1)&&(crntplane.NJDIRC==0)&&(crntplane.NKDIRC==0))
{
//vector transform into surf coordinates of n--100 plane
surfphx=phonon[7];
surfphy=phonon[6];
surfphz=phonon[5]*double(-1);
//vector transform into surf coordinates of n--100 plane
} else if ((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==0)&&(crntplane.NKDIRC==1))
{
//vector transform into surf coordinates of n-001 plane
surfphx=phonon[5];
surfphy=phonon[6];
surfphz=phonon[7];
//vector transform into surf coordinates of n-001 plane
} else if ((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==0)&&(crntplane.NKDIRC==--1))
{
//vector transform into surf coordinates of n-00-1 plane
surfphx=phonon[5];
surfphy=phonon[6]*double(-1);
surfphz=phonon[7]*double(-1);
//vector transform into surf coordinates of n-00-1 plane
} else
{
cout << "undefined surface plane" << endl;
system ("pause");
}
}

```

```

}

surfphxorg=surfphx;
surfphyorg=surfphy;
surfphzorg=surfphz;

//cout << "*****" << endl << "Surfsim converted phonon details:" << endl << surfphx
<< "*****" << surfphy << "*****" << surfphz << "*****" << endl;
n=0;
reenter=1;

while (reenter==1)
{
    reenter=0;
    rmatindx = randoms.RandInt(rrange, (511-rrange));
    rmatindy = randoms.RandInt(rrange, (511-rrange));
    n=n+1;

    nafm = regplnormvec(afmroughness, rmatindx, rmatindy, rrange);

    cout << "hit point on afm image: " << afmroughness[rmatindx][rmatindy] << endl;

    totafmcount++;

    if ((180.0*double(atan(nafm[2]/sqrt(double((nafm[0]*nafm[0])+(nafm[1]*nafm[1])))))/PI) <
60.0)
    {
        bel80count++;

        cout << "+++++++ Afm surf normal: " << nafm[0] << "; " << nafm[1] << "; " <<
nafm[2] << "; " << nafm[2]/sqrt(double((nafm[0]*nafm[0])+(nafm[1]*nafm[1])))<< endl;
        cout << "Surface normal angle: " <<
180.0*atan(nafm[2]/sqrt(double((nafm[0]*nafm[0])+(nafm[1]*nafm[1])))/PI << endl;
        cout << bel80count<< " out of " << totafmcount<< endl;

    }

    /*cout << "*****" << endl << "Surfsim non processed phonon details:" << endl <<
surfphx << "*****" << surfphy << "*****" << surfphz << "*****" << endl;

    cout << "+++++++ Afm surf normal: " << nafm[0] << "; " << nafm[1] << "; " <<
nafm[2] << "; " << endl;

    system("pause");
}*/

dotproln= (surfphx*nafm[0]) + (surfphy*nafm[1]) +(surfphz*nafm[2]);

if (dotproln > 0)
{
    //cout << "dotproln >0 case: " << dotproln << endl;
    reenter=1;
} else
{
    surfphx=surfphx -(2.0*dotproln*nafm[0]);
    surfphy=surfphy -(2.0*dotproln*nafm[1]);
    surfphz=surfphz -(2.0*dotproln*nafm[2]);

    if (surfphz<0.001)
    {
        reenter=1;
    }else if (dotproln!=dotproln)
    {

```

```

    cout << "Indefinite number issue in surfsim algorithm. doproIn is indefinite. Re-entering
the surface with original phonons"<< endl;
    system ("pause");
    surfphx=surfphxorg;
    surfphy=surfphyorg;
    surfphz=surfphzorg;
    reenter=1;
}

}

}

//cout << "*****"<< endl << "Surfsim processed phonon details:" << endl<< surfphx
<< "*****" << surfphy << "*****" << surfphz << "*****" <<endl;

if((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==1)&&(crrntplane.NKDIRC==0))
{
    // back vector transform into surf coordinates of n-010 plane
    phonon[5]=surfphx;
    phonon[7]=surfphy*double(-1);
    phonon[6]=surfphz;
    // back vector transform into surf coordinates of n-010 plane
}
else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==1)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n-0-10 plane
    phonon[5]=surfphx;
    phonon[7]=surfphy;
    phonon[6]=surfphz*double(-1);
    //vector transform into surf coordinates of n-0-10 plane
}
else if ((crrntplane.NIDIRC==1)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n-100 plane
    phonon[7]=surfphx*double(-1);
    phonon[6]=surfphy;
    phonon[5]=surfphz;
    //vector transform into surf coordinates of n-100 plane
} else if ((crrntplane.NIDIRC==1)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n--100 plane
    phonon[7]=surfphx;
    phonon[6]=surfphy;
    phonon[5]=surfphz*double(-1);
    //vector transform into surf coordinates of n--100 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
    //vector transform into surf coordinates of n-001 plane
    phonon[5]=surfphx;
    phonon[6]=surfphy;
    phonon[7]=surfphz;
    //vector transform into surf coordinates of n-001 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
    //vector transform into surf coordinates of n-00-1 plane
    surfphx=phonon[5];
    phonon[6]=surfphy*double(-1);
    phonon[7]=surfphz*double(-1);
    //vector transform into surf coordinates of n-00-1 plane
}

double phIvec = phonon[5];
double phJvec = phonon[6];
double phKvec = phonon[7];

```

```

//cout << "*****" << endl << "Surfsim exiting phonon details:" << endl << phonon[5]
<< "*****" << phonon[6] << "*****" << phonon[7] << "*****" << endl;
double dotprorn;
if (ifpffafm==1)
{
    ppolar=randoms.RandInt(1,3);
    pff(phIvec,phJvec,phKvec,phonon[0],ppolar); //turn on phonon focusing factors
    dotprorn= (phIvec*crntplane.NIDIRC) + (phJvec*crntplane.NJDIRC)
    +(phKvec*crntplane.NKDIRC);
    if (dotprorn <0)
    {
        phIvec=-phIvec;
        phJvec=-phJvec;
        phKvec=-phKvec;
    }
}
else
{
    dotprorn = (phIvec*crntplane.NIDIRC) + (phJvec*crntplane.NJDIRC)
    +(phKvec*crntplane.NKDIRC);

    if (dotprorn <0)
    {
        cout << "There is specrefl phonon reflection problem. The angle between the reflected phonon
        and the surface normal exceeds 90 degrees, ignoring phonon" << endl;
        system ("pause");
        //scatproblem=1;
    }
}

phonon[5] = phIvec;
phonon[6] = phJvec;
phonon[7] = phKvec;

//cout << "Number of surf afm bounces: " << n << endl;
}

// Advanced surface simulation algorithm trial. It tries to simulation phonon-surface
interactions based virtual conic features on the surface. This is not actively used in
nanosheet phonon transport MC simulations.
void surfsimcone(tvector<double> &phonon, PLANE crntplane)
{
    double surfphx, surfphy, surfphz,surfphxorg, surfphyorg, surfphzorg, zintcone, yintcone,
    xintcone, xintcone1, xintcone2,nconex,ncone,ncone3,dotproln;
    int repint, reenter, n;
    RandGen randoms;
    //cout << "*****" << endl << "Surfsim entering phonon details:" << endl << phonon[5]
    << "*****" << phonon[6] << "*****" << phonon[7] << "*****" << endl;
    if((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==1)&&(crntplane.NKDIRC==0))
    {
        //vector transform into surf coordinates of n-010 plane
        surfphx=phonon[5];
        surfphy=phonon[7]*double(-1);
        surfphz=phonon[6];
        //vector transform into surf coordinates of n-010 plane
    }
    else if ((crntplane.NIDIRC==0)&&(crntplane.NJDIRC==1)&&(crntplane.NKDIRC==0))
    {
        //vector transform into surf coordinates of n-0-10 plane
        surfphx=phonon[5];
        surfphy=phonon[7];
        surfphz=phonon[6]*double(-1);
        //vector transform into surf coordinates of n-0-10 plane
    }
    else if ((crntplane.NIDIRC==1)&&(crntplane.NJDIRC==0)&&(crntplane.NKDIRC==0))
    {
        //vector transform into surf coordinates of n-100 plane

```

```

surfphx=phonon[7]*double(-1);
surfphy=phonon[6];
surfphz=phonon[5];
//vector transform into surf coordinates of n-100 plane
} else if ((crrntplane.NIDIRC==-1)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==0))
{
//vector transform into surf coordinates of n--100 plane
surfphx=phonon[7];
surfphy=phonon[6];
surfphz=phonon[5]*double(-1);
//vector transform into surf coordinates of n--100 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
//vector transform into surf coordinates of n-001 plane
surfphx=phonon[5];
surfphy=phonon[6];
surfphz=phonon[7];
//vector transform into surf coordinates of n-001 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
//vector transform into surf coordinates of n-00-1 plane
surfphx=phonon[5];
surfphy=phonon[6]*double(-1);
surfphz=phonon[7]*double(-1);
//vector transform into surf coordinates of n-00-1 plane
} else
{

cout << "undefined surface plane" << endl;
system ("pause");
}

surfphxorg=surfphx;
surfphyorg=surfphy;
surfphzorg=surfphz;

//cout << "*****" << endl << "Surfsim converted phonon details:" << endl << surfphx
<< "*****" << surfphy << "*****" << surfphz << "*****" << endl;
n=0;
reenter=1;
repint=1;
while (reenter==1)
{
reenter=0;
zintcone = randoms.RandReal(0.1, 0.9999);
n=n+1;
while (repint==1)
{
repint=0;
yintcone = randoms.RandReal(double(-1.0)*sqrt(double((zintcone-double(1.0))*(zintcone-
double(1.0)))),-sqrt(double((zintcone-double(1.0))*(zintcone-double(1.0)))));
xintcone1 = sqrt(double(double((zintcone-double(1.0))*(zintcone-double(1.0))) -
double(yintcone*yintcone)));
xintcone2 = double(-1.0)*double(sqrt(double(double((zintcone-double(1.0))*(zintcone-
double(1.0))) - double(yintcone*yintcone))));

if (((surfphx < 0)&&(surfphy>0))||((surfphx > 0)&&(surfphy<0)))
{
if ((((-1.0*surfphx)*yintcone) < (surfphy*xintcone1))||(((-1.0*surfphx)*yintcone) <
(surfphy*xintcone2)))
{
if ((((-1.0*surfphx)*yintcone) < (surfphy*xintcone1))&&(((-1.0*surfphx)*yintcone) <
(surfphy*xintcone2)))
{
if (randoms.RandReal(0,1)<0.5)
{
xintcone=xintcone1;

```

```

    }else
    {
        xintcone=xintcone2;
    }
    } else if (((-1.0*surfphx)*yintcone) < (surfphy*xintcone1))
    {
        xintcone=xintcone1;
    }else
    {
        xintcone=xintcone2;
    }

    }else
    {
        //cout<< "-----cone could not intesect 1" << endl << "---*---" << xintcone1 <<"-----
        -" << xintcone2<<"---*---" << yintcone <<"-----" << zintcone;
        repint==1;
    }
    } else if (((surfphx < 0)&&(surfphy < 0)) || ((surfphx > 0)&&(surfphy > 0)))
    {
        if ((((-1.0*surfphx)*yintcone) > (surfphy*xintcone1))||(((-1.0*surfphx)*yintcone) >
(surfphy*xintcone2)))
        {
            if ((((-1.0*surfphx)*yintcone) > (surfphy*xintcone1))&&(((-1.0*surfphx)*yintcone) >
(surfphy*xintcone2)))
            {
                if (randoms.RandReal(0,1)<0.5)
                {
                    xintcone=xintcone1;
                }else
                {
                    xintcone=xintcone2;
                }
            } else if (((-1.0*surfphx)*yintcone) > (surfphy*xintcone1))
            {
                xintcone=xintcone1;
            }else
            {
                xintcone=xintcone2;
            }
        }
    }else
    {
        //cout<< "-----cone could not intesect 2" << endl << "---*---" << xintcone1 <<"-----
        -" << xintcone2<<"---*---" << yintcone <<"-----" << zintcone;
        repint==1;
    }
    }
}

//cout << "*****"<< endl << "Surfsim cone intersection details:" << endl<<
xintcone << "*****" << yintcone << "*****" << zintcone << "*****" <<endl;

nconex=double(2.0)*xintcone;
nconey=double(2.0)*yintcone;
nconez=double(-2.0)*zintcone;

dotproln= (surfphx*nconex) + (surfphy*nconey) +(surfphz*nconez);

surfphx=surfphx -(2.0*dotproln*nconex);
surfphy=surfphy -(2.0*dotproln*nconey);
surfphz=surfphz -(2.0*dotproln*nconez);

if (surfphz<0.001)
{

```

```

    reenter=1;
}else if (dotproln!=dotproln)
{
    cout << "Indefinite number issue in surfsim algorithm. doproln is indefinite. Re-entering the
surface with original phonons"<< endl;
    system ("pause");
    surfphx=surfphxorg;
    surfphy=surfphyorg;
    surfphz=surfphzorg;
    reenter=1;
}

}
//cout << "*****"<< endl << "Surfsim processed phonon details:" << endl<< surfphx
<< "*****" << surfphy << "*****" << surfphz << "*****" <<endl;

if((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==1)&&(crrntplane.NKDIRC==0))
{
    // back vector transform into surf coordinates of n-010 plane
    phonon[5]=surfphx;
    phonon[7]=surfphy*double(-1);
    phonon[6]=surfphz;
    // back vector transform into surf coordinates of n-010 plane
}
else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==1)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n-0-10 plane
    phonon[5]=surfphx;
    phonon[7]=surfphy;
    phonon[6]=surfphz*double(-1);
    //vector transform into surf coordinates of n-0-10 plane
}
else if ((crrntplane.NIDIRC==1)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n-100 plane
    phonon[7]=surfphx*double(-1);
    phonon[6]=surfphy;
    phonon[5]=surfphz;
    //vector transform into surf coordinates of n-100 plane
} else if ((crrntplane.NIDIRC==1)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==0))
{
    //vector transform into surf coordinates of n--100 plane
    phonon[7]=surfphx;
    phonon[6]=surfphy;
    phonon[5]=surfphz*double(-1);
    //vector transform into surf coordinates of n--100 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
    //vector transform into surf coordinates of n-001 plane
    phonon[5]=surfphx;
    phonon[6]=surfphy;
    phonon[7]=surfphz;
    //vector transform into surf coordinates of n-001 plane
} else if ((crrntplane.NIDIRC==0)&&(crrntplane.NJDIRC==0)&&(crrntplane.NKDIRC==1))
{
    //vector transform into surf coordinates of n-00-1 plane
    surfphx=phonon[5];
    phonon[6]=surfphy*double(-1);
    phonon[7]=surfphz*double(-1);
    //vector transform into surf coordinates of n-00-1 plane
}

//cout << "*****"<< endl << "Surfsim exiting phonon details:" << endl<< phonon[5]
<< "*****" << phonon[6] << "*****" << phonon[7] << "*****" <<endl;

double dotprorn= (phonon[5]*crrntplane.NIDIRC) + (phonon[6]*crrntplane.NJDIRC)
+(phonon[7]*crrntplane.NKDIRC);

```



```

if (dotprorn < 0)
{
    cout << "There is specrefl phonon reflection problem. The angle between the reflected phonon
and the surface normal exceeds 90 degrees, ignoring phonon" << endl;
    system ("pause");
    //scatproblem=1;
}

cout << "Number of surf afm bounces: " << n << endl;

}

// function for transforming incident phonon to reflected phonon
void specrefl(tvector<double> &phonon, PLANE crntplane, int &scatproblem)
{
    double dotproln= (phonon[5]*crntplane.NIDIRC) + (phonon[6]*crntplane.NJDIRC)
+(phonon[7]*crntplane.NKDIRC); //dot product of phonon vector and plane vectors
    if (dotproln!=dotproln)
    {
        /*cout << "Indefinite number issue in specrefl algorithm. dotproln is indefinite"<< endl;
        system ("pause");*/
    }
    phonon[5]= phonon[5] -(2.0*dotproln*crntplane.NIDIRC);
    phonon[6]= phonon[6] -(2.0*dotproln*crntplane.NJDIRC);
    phonon[7]= phonon[7] -(2.0*dotproln*crntplane.NKDIRC);
    /*cout << "Specularly Reflected" << endl;
    cout << "frequency " << phonon[0] << "phonon referencenumber " << phonon[1] << "x initial
position" << phonon[2] << "y initial position" << phonon[3] << "z initial position" <<
phonon[4] << "i vector" << phonon[5] << "j vector" << phonon[6] << "k vector" << phonon[7]
<<endl;
    system ("pause");*/
    double dotprorn= (phonon[5]*crntplane.NIDIRC) + (phonon[6]*crntplane.NJDIRC)
+(phonon[7]*crntplane.NKDIRC);
    if (dotprorn < 0)
    {
        cout << "There is specrefl phonon reflection problem. The angle between the reflected phonon
and the surface normal exceeds 90 degrees, ignoring phonon" << endl;
        //system ("pause");
        scatproblem=1;
    }
}

// function for transforming incident phonon to randomly scattered phonon. New phonon
direction can be selected from cosine dist. (ifcos) and pff can be taken account for scattered
phonons
void diffscatt(tvector<double> &phonon, PLANE crntplane, int ifcos, int &scatproblem, int
ifpffscatt)
{
    RandGen randoms;
    int ppolar;
    tvector<double> randveconpl(0);
    tvector<double> secveconpl(0);
    tvector<double> temprandvec(0);
    genrandveconpl(randveconpl,crntplane);
    secveconpl.push_back((crntplane.NJDIRC*randveconpl[2])-(crntplane.NKDIRC*randveconpl[1]));
    secveconpl.push_back((crntplane.NKDIRC*randveconpl[0])-(crntplane.NIDIRC*randveconpl[2]));
    secveconpl.push_back((crntplane.NIDIRC*randveconpl[1])-(crntplane.NJDIRC*randveconpl[0]));
    /*cout << "secondary vector on plane magnitude " << endl;
    vecmag(secveconpl);*/
    /*cout << endl;*/
    genrandvec(temprandvec,0,360,ifcos);
    double
    phIvec=(crntplane.NIDIRC*temprandvec[2])+(randveconpl[0]*temprandvec[0])+(secveconpl[0]*tempr
andvec[1]);
    double
    phJvec=(crntplane.NJDIRC*temprandvec[2])+(randveconpl[1]*temprandvec[0])+(secveconpl[1]*tempr
andvec[1]);

```

```

double
phKvec=(crrntplane.NKDIRC*temprandvec[2])+(randveconpl[2]*temprandvec[0])+(secveconpl[2]*tempr
andvec[1]);
if (phIvec!=phIvec)
{
/*cout << "Indefinite number issue in diffscatt algorithm. PhIvec is indefinite"<< endl;
system ("pause");*/
}
if (phJvec!=phJvec)
{
/*cout << "Indefinite number issue in diffscatt algorithm. PhJvec is indefinite"<< endl;
system ("pause");*/
}
if (phKvec!=phKvec)
{
/*cout << "Indefinite number issue in diffscatt algorithm. PhKvec is indefinite"<< endl;
system ("pause");*/
}
//system ("pause");
if (ifpffscatt==1)
{
ppolar=randoms.RandInt(1,3); //send polarization info to pff function
pff(phIvec,phJvec,phKvec,phonon[0],ppolar); //turn on phonon focusing factors
double dotprosn= (phIvec*crrntplane.NIDIRC) + (phJvec*crrntplane.NJDIRC)
+(phKvec*crrntplane.NKDIRC);
if (dotprosn <0)
{
phIvec=-phIvec;
phJvec=-phJvec;
phKvec=-phKvec;
}
} else
{
double dotprosn= (phIvec*crrntplane.NIDIRC) + (phJvec*crrntplane.NJDIRC)
+(phKvec*crrntplane.NKDIRC);
if (dotprosn <0)
{
cout << "There is diffscatt phonon scattering problem. The angle between the scattering
phonon and the surface normal exceeds 90 degrees, ignoring phonon" << endl;
//system ("pause");
scatproblem=1;
}
}

phonon[5]=(phIvec);
phonon[6]=(phJvec);
phonon[7]=(phKvec);
/*phvecmag(phonon);*/
/*cout << "Diffusively scattered" << endl;*/
//cout << "frequency " << phonon[0] << "phonon referencenumber " << phonon[1] << "x initial
position" << phonon[2] << "y initial position" << phonon[3] << "z initial position" <<
phonon[4] << "i vector" << phonon[5] << "j vector" << phonon[6] << "k vector" << phonon[7]
<<endl;
/*system ("pause");*/

}

//function for generating phonons at the generator plane, source can be selected to be plane
and point source, pff can be taken into account in phonon generation
void generatepls(tvector<double> &phonon, const tvector<PLANE> &planelist, double frequency,
double phrefnum, int ifpsource, int ifcos, int ifpffgen)
{
phonon[0]=(frequency);
phonon[1]=(phrefnum);
int ppolar;

```

```

RandGen randoms;
tvector<double> randveconpl(0);
tvector<double> secveconpl(0);
tvector<double> temprandvec(0);
double ptxpos,ptypos, ptzpos;
if (ifpsource==1)
{
    ptxpos = (planelist[4].XLIMHIGH+planelist[4].XLIMLOW)/2;
    ptypos = (planelist[4].YLIMHIGH+planelist[4].YLIMLOW)/2;
    if (planelist[4].ZPARAM==0)
    {
        ptzpos = (planelist[4].ZLIMHIGH+planelist[4].ZLIMLOW)/2;
    }
    else
    {
        ptzpos = (planelist[4].PCONST-(planelist[4].XPARAM*ptxpos)-
(planelist[4].YPARAM*ptypos))/planelist[4].ZPARAM;
    }
}
else
{
    if (planelist[4].XLIMLOW == planelist[4].XLIMHIGH)
    {
        ptxpos = planelist[4].XLIMHIGH;
    }
    else
    {
        ptxpos = randoms.RandReal(planelist[4].XLIMLOW,planelist[4].XLIMHIGH);
        if (ptxpos==planelist[4].XLIMLOW || ptxpos==planelist[4].XLIMHIGH)
        {
            cout <<"Random x position at intersect: " << ptxpos<<endl;
        }
    }
    if (planelist[4].YLIMLOW == planelist[4].YLIMHIGH)
    {
        ptypos = planelist[4].YLIMHIGH;
    }
    else
    {
        ptypos = randoms.RandReal(planelist[4].YLIMLOW,planelist[4].YLIMHIGH);
    }
    if (planelist[4].ZPARAM==0)
    {
        ptzpos = randoms.RandReal(planelist[4].ZLIMLOW,planelist[4].ZLIMHIGH);
        if (ptzpos==planelist[4].ZLIMLOW || ptzpos==planelist[4].ZLIMHIGH)
        {
            cout <<"Random z position at intersect: " << ptzpos<<endl;
        }
    }
    else
    {
        ptzpos = (planelist[4].PCONST-(planelist[4].XPARAM*ptxpos)-
(planelist[4].YPARAM*ptypos))/planelist[4].ZPARAM;
    }
}

if (ptxpos!=ptxpos)
{
    /*cout << "Indefinite number issue in generatepls algorithm. ptxpos is indefinite"<< endl;
    system ("pause");*/
}
if (ptypos!=ptypos)
{
    /*cout << "Indefinite number issue in generatepls algorithm. ptxpos is indefinite"<< endl;
    system ("pause");*/
}

```

```

}
if (ptzpos!=ptzpos)
{
/*cout << "Indefinite number issue in generatepls algorithm. ptxpos is indefinite"<< endl;
system ("pause");*/
}
phonon[2]=(ptxpos);
phonon[3]=(ptypos);
phonon[4]=(ptzpos);
genrandveconpl(randveconpl,planelist[4]);
secveconpl.push_back((planelist[4].NJDIRC*randveconpl[2]))-
(planelist[4].NKDIRC*randveconpl[1]));
secveconpl.push_back((planelist[4].NKDIRC*randveconpl[0]))-
(planelist[4].NJDIRC*randveconpl[2]));
secveconpl.push_back((planelist[4].NJDIRC*randveconpl[1]))-
(planelist[4].NJDIRC*randveconpl[0]));
/*cout << "secondary vector on plane magnitude " << endl;
vecmag(secveconpl);*/
/*cout << endl;*/
genrandvec(temprandvec,0,360,ifcos); /*for general use*/
double
phIvec=(planelist[4].NJDIRC*temprandvec[2])+(randveconpl[0]*temprandvec[0])+(secveconpl[0]*tem
prandvec[1]);
double
phJvec=(planelist[4].NJDIRC*temprandvec[2])+(randveconpl[1]*temprandvec[0])+(secveconpl[1]*tem
prandvec[1]);
double
phKvec=(planelist[4].NKDIRC*temprandvec[2])+(randveconpl[2]*temprandvec[0])+(secveconpl[2]*tem
prandvec[1]);
if (phIvec!=phIvec)
{
/*cout << "Indefinite number issue in generatepls algorithm. PhIvec is indefinite"<< endl;
system ("pause");*/
}
if (phJvec!=phJvec)
{
/*cout << "Indefinite number issue in generatepls algorithm. PhJvec is indefinite"<< endl;
system ("pause");*/
}
if (phKvec!=phKvec)
{
/*cout << "Indefinite number issue in generatepls algorithm. PhKvec is indefinite"<< endl;
system ("pause");*/
}
if (ifpffgen==1)
{
polar=randoms.RandInt(1,3);
pff(phIvec,phJvec,phKvec,phonon[0],ppolar); //turn on phonon focusing factors
double dotprosn= (phIvec*planelist[4].NJDIRC) + (phJvec*planelist[4].NJDIRC)
+(phKvec*planelist[4].NKDIRC);
//system ("pause");
if (dotprosn <0)
{
phIvec=-phIvec;
phJvec=-phJvec;
phKvec=-phKvec;
}
}
phonon[5]=(phIvec);
phonon[6]=(phJvec);
phonon[7]=(phKvec);
/*phvecmag(phonon);*/
/*cout<< "Generated Phonon"<<endl;*/
//cout << "frequency " << phonon[0] << "phonon referencenumber " << phonon[1] << "x initial
position" << phonon[2] << "y initial position" << phonon[3] << "z initial position" <<
phonon[4] << "i vector" << phonon[5] << "j vector" << phonon[6] << "k vector" << phonon[7]
<<endl;

```

```

    //system ("pause");
}

//this function determines which plane that the phonon intersects with. The infinite line
along the phonon direction intersects with many planes.
bool ifintersect(double &t,const tvector<double> &phonon, PLANE crrntplane, int recplindmem)
//, string &intreport,string &intreportlucky
{
    bool ifint=0;
    double ttemp=0;

    /*cout << "intersection function run" << endl;*/
    if
    (((crrntplane.XPARAM*phonon[5])+(crrntplane.YPARAM*phonon[6])+(crrntplane.ZPARAM*phonon[7]))
    == 0)
    {
        ifint=0;
        //cout << "Phonon goes parallel with " << crrntplane.PLADISC << "Phonon Vectors: " <<
        phonon[5]<< ", " << phonon[6] << ", " << phonon[7]<< endl;

    }
    else
    {
        ttemp = (crrntplane.PCONST-
        ((crrntplane.XPARAM*phonon[2])+(crrntplane.YPARAM*phonon[3])+(crrntplane.ZPARAM*phonon[4])))/
        ((crrntplane.XPARAM*phonon[5])+(crrntplane.YPARAM*phonon[6])+(crrntplane.ZPARAM*phonon[7]));
        double ptxposnew=(ttemp*phonon[5])+phonon[2];
        double ptyposnew=(ttemp*phonon[6])+phonon[3];
        double ptzposnew=(ttemp*phonon[7])+phonon[4];
        if (ptxposnew!=ptxposnew)
        {
            /*cout << "Indefinite number issue in ifintersect algorithm. ptxposnew is indefinite"<< endl;
            system ("pause");*/
        }
        if (ptyposnew!=ptyposnew)
        {
            /*cout << "Indefinite number issue in ifintersect algorithm. ptyposnew is indefinite"<< endl;
            system ("pause");*/
        }
        if (ptzposnew!=ptzposnew)
        {
            /*cout << "Indefinite number issue in ifintersect algorithm. ptzposnew is indefinite"<< endl;
            system ("pause");*/
        }
        if (ttemp!=ttemp)
        {
            /*cout << "Indefinite number issue in ifintersect algorithm. ttemp is indefinite"<< endl;
            system ("pause");*/
        }
        /*cout << "Phonon with frequency " << phonon[0] << "phonon referencenumber " << phonon[1] <<
        "x initial position" << phonon[2] << "y initial position" << phonon[3] << "z initial position"
        << phonon[4] << "i vector" << phonon[5] << "j vector" << phonon[6] << "k vector" << phonon[7]
        <<endl;
        cout <<"Imaginary intersection with "<< crrntplane.PLADISC << " at " << ptxposnew << ", " <<
        ptyposnew << ", " << ptzposnew << endl;*/
        //intreport=intreport+"***. Intersection with "+ crrntplane.PLADISC + " at x=" +
        toString(double(ptxposnew))+ " at y=" + toString(double(ptyposnew))+ " at z=" +
        toString(double(ptzposnew))+ ". Temporary t value"+ toString(double(ttemp));

        if (recplindmem==crrntplane.REFNUM)
        {
            /*cout <<"Intesection with the emerging plane." <<endl;*/
            ttemp=0;
        }
        if (crrntplane.XLIMHIGH == crrntplane.XLIMLOW)
        {

```

```

    if (ptxposnew != crrentplane.XLIMHIGH)
    {
        /*cout << "Floating point issue. Rounding numbers by:" << ptxposnew-crrntplane.XLIMHIGH <<
endl;*/
        ptxposnew=crrntplane.XLIMHIGH;
    }

} else if (crrntplane.YLIMHIGH == crrntplane.YLIMLOW)
{
    if (ptyposnew != crrntplane.YLIMHIGH)
    {
        /*cout << "Floating point issue. Rounding numbers." << ptyposnew-crrntplane.YLIMHIGH <<
endl;*/
        ptyposnew=crrntplane.YLIMHIGH;
    }

} else if (crrntplane.ZLIMHIGH == crrntplane.ZLIMLOW)
{
    if (ptzposnew != crrntplane.ZLIMHIGH)
    {
        /*cout << "Floating point issue. Rounding numbers." << ptzposnew-crrntplane.ZLIMHIGH <<
endl;*/
        ptzposnew=crrntplane.ZLIMHIGH;
    }
}
if (((ptxposnew >= crrntplane.XLIMLOW)&&(ptxposnew <= crrntplane.XLIMHIGH))&&((ptyposnew >=
crrntplane.YLIMLOW)&&(ptyposnew <= crrntplane.YLIMHIGH))&&((ptzposnew >=
crrntplane.ZLIMLOW)&&(ptzposnew <= crrntplane.ZLIMHIGH)))
{

    ifint=1;
    t=ttemp;

    /*cout << "YES intesects with " << crrntplane.PLADISC << " with t= " << t << endl;*/
    /*system("pause");*/
    //intreportlucky=intreportlucky+****. Intersection with "+ crrntplane.PLADISC + " at x=" +
toString(double(ptxposnew))+ " at y=" + toString(double(ptyposnew))+ " at z=" +
toString(double(ptzposnew));

}
else
{
    /*if (((ptxposnew >= (crrntplane.XLIMLOW-0.000001))&&(ptxposnew <=
(crrntplane.XLIMHIGH+0.000001)))&&((ptyposnew >= (crrntplane.YLIMLOW-0.000001))&&(ptyposnew <=
(crrntplane.YLIMHIGH+0.000001)))&&((ptzposnew >= (crrntplane.ZLIMLOW-0.000001))&&(ptzposnew <=
(crrntplane.ZLIMHIGH+0.000001))))
    {
        cout << "Well... It didn't make it but " << crrntplane.PLADISC << " was pretty close"
<<endl;
    }*/

    ifint=0;

    /*
cout << "Phonon intersects the plane " << crrntplane.PLADISC <<" out of boundaries" << endl;
if ((ptxposnew < crrntplane.XLIMLOW) || (ptxposnew > crrntplane.XLIMHIGH))
{
    cout << "X-limit exceeded" <<crrntplane.XLIMLOW<< "," <<crrntplane.XLIMHIGH<< endl;
} else if ((ptyposnew < crrntplane.YLIMLOW)|| (ptyposnew > crrntplane.YLIMHIGH))
{
    cout << "Y-limit exceeded" <<crrntplane.YLIMLOW<< "," <<crrntplane.YLIMHIGH<< endl;
} else if ((ptzposnew < crrntplane.ZLIMLOW)|| (ptzposnew > crrntplane.ZLIMHIGH))
{
    cout << "Z-limit exceeded" <<crrntplane.ZLIMLOW<< "," <<crrntplane.ZLIMHIGH<< endl;
}
}

```

```

    }
    else
    {
        cout << "There is something wrong"<< endl;
    }*/

}
}
return ifint;
}

//This function generates planes for the nanosheets, mesa sidewalls and system boundaries.
some planes are assigned as generator or detector.
void generateplanes(tvector<PLANE> &planelist, int howmanyys, double nspos, double nslength,
double nswidth, double nspitch, double nsdepth, double mesawidth, double mesaheight, double
nstodet, double mesatoptobot, double detwidth, double genwidth, double xposborder, double
xnegborder, double yposborder, double ynegborder, double nsrough, double msrough, double
mtrough, double wbrough, double nsgapbotrough, int fbotype, double detshift, double detratio,
int ifangrep)
{
    int pushind, pushind2, pushind3;
    pushind=0;
    pushind2=0;
    if (ifangrep==1)
    {
        pushind=2;
        pushind2=4;
    }

    double mesabotwidth = (2*(mesaheight/tan(double(54.75*PI/180))))+mesawidth;
    double mstilt=tan(double(54.75*PI/180));

    // define number of nanosheets. can't be even because of symmetry
    if ((howmanyys % 2 == 0) && (howmanyys != 0))
    {
        cout << "Number of nanosheets can't be even. Adding +1 to number of nanosheets"<< endl;
        howmanyys= howmanyys+1;
    }
    //check if the entered inputs geometrically make sense.
    if (xposborder < (nspos + (((howmanyys)*nspitch)+(nspitch-nswidth))/2)) || -xnegborder >
    (nspos - (((howmanyys)*nspitch)+(nspitch-nswidth))/2))
    {
        cout << "Your nanosheets are reaching out of system borders. Enlarging system borders to fit
all nanosheets plus a nanosheet pitch"<<endl;
        xposborder= (nspos + (((howmanyys)*nspitch)+(nspitch-nswidth))/2)+(nspitch/2);
        xnegborder= -(nspos - (((howmanyys)*nspitch)+(nspitch-nswidth))/2)-(nspitch/2));
    }

    if (((detwidth/2) > xposborder) || ((-detwidth/2) < (-xnegborder)))
    {
        cout << "Your detector width can not be wider than x system borders. Enlarging x system
borders to %150 of detector size..." << endl;
        xposborder=((detwidth/2)*3)/2;
        xnegborder=((detwidth/2)*3)/2;
    }

    if (((genwidth/2) > xposborder) || ((-genwidth/2) < (-xnegborder)))
    {
        cout << "Your generator width can not be wider than x system borders. Enlarging x system
borders to %150 of detector size..." << endl;
        xposborder=((genwidth/2)*3)/2;
        xnegborder=((genwidth/2)*3)/2;
    }
    if ( ynegborder < (mesabotwidth/2) || yposborder < (mesabotwidth/2) )
    {

```

```

    cout << "Your mesa width at the bottom of the mesa can not be wider than y system borders.
    Enlarging y system borders to %150 of mesa bottom width..." << endl;
    yposborder=((mesabotwidth/2)*3)/2;
    ynegborder=((mesabotwidth/2)*3)/2;
}

// define the very first nanosheet planes according to PLANE struct
planelist.push_back(newplane(0, "positive x system border plane", xposborder, xposborder, -
2*ynegborder, 2*yposborder, -mesatoptobot, 0, 1, 0, 0, xposborder, -1, 0, 0, 3,0,0));
planelist.push_back(newplane(1, "negative x system border plane", -xnegborder, -xnegborder, -
2*ynegborder, 2*yposborder, -mesatoptobot, 0, 1, 0, 0, -xnegborder, 1, 0, 0, 3,0,0));
planelist.push_back(newplane(2, "positive y system border plane", -2*xnegborder,
2*xposborder, yposborder, yposborder, -mesatoptobot, 0, 0, 1, 0, yposborder, 0, -1, 0,3,0,0));
planelist.push_back(newplane(3, "negative y system border plane", -2*xnegborder,
2*xposborder, -ynegborder, -ynegborder, -mesatoptobot, 0, 0, 1, 0, -ynegborder, 0, 1,
0,3,0,0));
planelist.push_back(newplane(4, "generator plane at mesa side wall", -(genwidth/2),
(genwidth/2), -(mesabotwidth/2), -(mesawidth/2), -mesaheight, 0, 0, -mstilt, 1,
(mesawidth/2)*mstilt, 0, mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)),1,msrough,0));
planelist.push_back(newplane(5, "non-generator positive x plane at mesa side wall",
(genwidth/2), xposborder, -(mesabotwidth/2), -(mesawidth/2), -mesaheight, 0, 0, -mstilt, 1,
(mesawidth/2)*mstilt, 0, mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)),1,msrough,0));
planelist.push_back(newplane(6, "non-generator negative x plane at mesa side wall", -
xnegborder, -(genwidth/2), -(mesabotwidth/2), -(mesawidth/2), -mesaheight, 0, 0, -mstilt, 1,
(mesawidth/2)*mstilt, 0, mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)),1,msrough,0));
planelist.push_back(newplane(7, "detector plane at mesa side wall", detshift-(detwidth/2),
detshift+(detwidth/2), (mesabotwidth/2)-(detratio*((mesabotwidth/2)-(mesawidth/2))),
(mesabotwidth/2), -mesaheight, mesaheight*(detratio-1), 0, mstilt, 1, (mesawidth/2)*mstilt, 0,
-mstilt/sqrt(double((mstilt*mstilt)+1)), -1/sqrt(double((mstilt*mstilt)+1)), 2,msrough,0));
planelist.push_back(newplane(8, "non-detector positive x plane at mesa side wall",
detshift+(detwidth/2), xposborder, (mesawidth/2), (mesabotwidth/2), -mesaheight, 0, 0, mstilt,
1, (mesawidth/2)*mstilt, 0, -mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)),1,msrough,0));
planelist.push_back(newplane(9, "non-detector negative x plane at mesa side wall", -
xnegborder,detshift-(detwidth/2), (mesawidth/2), (mesabotwidth/2), -mesaheight, 0, 0, mstilt,
1, (mesawidth/2)*mstilt, 0, -mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)),1,msrough,0));

double nsylimlow=(-mesawidth/2)+nstodet;
double nsylimhigh=nsylimlow+nslength;

// define the other nanosheet planes
int k=0;
for (k=0; k<=howmanyns-1; k+=2)
{

//if there is only one nanosheet the condition below works

if (k==0)
{

string kstr=itoa(k+1);
string kp1str=itoa(k+2);
string kp2str=itoa(k+3);
string kp3str=itoa(k+4);
string p1mod12="low x border nanosheet "+ kstr;
string p2mod12="high x border nanosheet "+ kstr;
string p3mod12="high x border nanosheet "+ kp1str;
string p4mod12="low x border nanosheet "+ kp2str;

string p5mod12="low y border of the gap bewteen nanosheet "+ kstr+" and "+ kp1str;
string p6mod12="high y border of the gap bewteen nanosheet "+ kstr+" and "+ kp1str;
string p7mod12="low y border of the gap bewteen nanosheet "+ kstr+" and "+ kp2str;
string p8mod12="high y border of the gap bewteen nanosheet "+ kstr+" and "+ kp2str;

```



```

string p9mod12="top plane of the nanosheet "+kstr;
string p10mod12="bottom plane of the nanosheet "+kstr;

string p11mod12="bottom plane of the gap between nanosheet " +kstr+" and "+kp1str;
string p12mod12="bottom plane of the gap between nanosheet " +kstr+" and "+kp2str;

    planelist.push_back(newplane(10, p1mod12 , nspos-(nswidth/2), nspos-(nswidth/2), nsylimlow,
    nsylimhigh, -nsdepth, 0, 1, 0, 0, nspos-(nswidth/2), 1, 0, 0,1,nsrrough,k+1)); //type should
    be 1
    planelist.push_back(newplane(11, p2mod12 , nspos+(nswidth/2), nspos+(nswidth/2), nsylimlow,
    nsylimhigh, -nsdepth, 0, 1, 0, 0, nspos+(nswidth/2), -1, 0, 0,1,nsrrough,k+1)); //type should
    be 1
    planelist.push_back(newplane(12, p3mod12 , nspos-(nspitch-nswidth/2), nspos-(nspitch-
    nswidth/2), nsylimlow, nsylimhigh, -nsdepth, 0, 1, 0, 0, nspos-(nspitch-nswidth/2), -1, 0,
    0,1,nsrrough,k+2)); //type should be 1
    planelist.push_back(newplane(13, p4mod12 , nspos+(nspitch-nswidth/2), nspos+(nspitch-
    nswidth/2), nsylimlow, nsylimhigh, -nsdepth, 0, 1, 0, 0, nspos+(nspitch-nswidth/2), 1, 0,
    0,1,nsrrough,k+3)); //type should be 1

    planelist.push_back(newplane(14, p5mod12 , nspos-(nspitch - nswidth/2), nspos-(nswidth/2),
    nsylimlow, nsylimlow, -nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,1,nsrrough,-1)); //type should
    be 1
    planelist.push_back(newplane(15, p6mod12 , nspos-(nspitch - nswidth/2), nspos-(nswidth/2),
    nsylimhigh, nsylimhigh, -nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,1,nsrrough,-1)); //type
    should be 1
    planelist.push_back(newplane(16, p7mod12 , nspos+(nswidth/2), nspos+(nspitch - nswidth/2),
    nsylimlow, nsylimlow, -nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,1,nsrrough,-1)); //type should
    be 1
    planelist.push_back(newplane(17, p8mod12 , nspos+(nswidth/2), nspos+(nspitch - nswidth/2),
    nsylimhigh, nsylimhigh, -nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,1,nsrrough,-1)); //type
    should be 1

    planelist.push_back(newplane(18, p9mod12 , nspos-(nswidth/2), nspos+(nswidth/2), nsylimlow,
    nsylimhigh, 0, 0, 0, 0, 1, 0, 0, 0, -1,1,mtrough,k+1)); //normally mtrough //type should be 1
    planelist.push_back(newplane(19, p10mod12 , nspos-(nswidth/2), nspos+(nswidth/2), nsylimlow,
    nsylimhigh, -nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, 1,5,nsrrough,k+1)); //normally regsys
    5
    planelist.push_back(newplane(20, p11mod12 , nspos-(nspitch - nswidth/2), nspos-(nswidth/2),
    nsylimlow, nsylimhigh, -nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, -1,1,nsrrough,-1));
    //type should be 1
    planelist.push_back(newplane(21, p12mod12 , nspos+(nswidth/2), nspos+(nspitch - nswidth/2),
    nsylimlow, nsylimhigh, -nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, -1,1,nsrrough,-1));
    //type should be 1

    // place imaginarry planes to record entering exiting phonon details
    if (ifangrep==1)
    {
        string angrep1="imaginery entrance plane of the nanosheet "+kstr;
        string angrep2="imaginery exit plane of the nanosheet "+kstr;

        pushind=2;

        planelist.push_back(newplane(22, angrep1 , nspos-(nswidth/2), nspos+(nswidth/2), nsylimlow,
        nsylimlow, -nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,4,nsrrough,k+1));
        planelist.push_back(newplane(23, angrep2 , nspos-(nswidth/2), nspos+(nswidth/2), nsylimhigh,
        nsylimhigh, -nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,4,nsrrough,k+1));
    }

    }
    else // this is the case with many nanosheets
    {
        string kstr=itoa(k);
        string kp1str=itoa(k+1);
        string kp2str=itoa(k+2);
        string kp3str=itoa(k+3);
    }

```

```

string p1mod14="low x border nanosheet "+ kstr;
string p2mod14="high x border nanosheet "+ kp1str;
string p3mod14="high x border nanosheet "+ kp2str;
string p4mod14="low x border nanosheet "+ kp3str;

string p5mod14="low y border of the gap bewteen nanosheet "+ kstr+" and "+ kp2str;
string p6mod14="high y border of the gap bewteen nanosheet "+ kstr+" and "+ kp2str;
string p7mod14="low y border of the gap bewteen nanosheet "+ kp1str+" and "+ kp3str;
string p8mod14="high y border of the gap bewteen nanosheet "+ kp1str+" and "+ kp3str;

string p9mod14="top plane of the nanosheet "+kstr;
string p10mod14="top plane of the nanosheet "+kp1str;
string p11mod14="bottom plane of the nanosheet "+kstr;
string p12mod14="bottom plane of the nanosheet "+kp1str;
string p13mod14="bottom plane of the gap between nanosheet " +kstr+" and "+kp2str;
string p14mod14="bottom plane of the gap between nanosheet " +kp1str+" and "+kp3str;

    planelist.push_back(newplane(pushind+22+(((k/2)-1)*(14+pushind2))), p1mod14 , nspos-
(nswidth/2)-((k/2)*nspitch), nspos-(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimhigh, -
nsdepth, 0, 1, 0, 0, nspos-(nswidth/2)-((k/2)*nspitch), 1, 0, 0,1,nsrough,k)); //type should
be 1
    planelist.push_back(newplane(pushind+23+(((k/2)-1)*(14+pushind2))), p2mod14 ,
nspos+(nswidth/2)+((k/2)*nspitch), nspos+(nswidth/2)+((k/2)*nspitch), nsylimlow, nsylimhigh, -
nsdepth, 0, 1, 0, 0, nspos+(nswidth/2)+((k/2)*nspitch), -1, 0, 0,1,nsrough,k+1)); //type
should be 1
    planelist.push_back(newplane(pushind+24+(((k/2)-1)*(14+pushind2))), p3mod14 , nspos-(nspitch-
nswidth/2)-((k/2)*nspitch), nspos-(nspitch-nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimhigh,
nsdepth, 0, 1, 0, 0, nspos-(nspitch-nswidth/2)-((k/2)*nspitch), -1, 0, 0,1,nsrough,k+2));
//type should be 1
    planelist.push_back(newplane(pushind+25+(((k/2)-1)*(14+pushind2))), p4mod14 , nspos+(nspitch-
nswidth/2)+((k/2)*nspitch), nspos+(nspitch-nswidth/2)+((k/2)*nspitch), nsylimlow, nsylimhigh,
nsdepth, 0, 1, 0, 0, nspos+(nspitch-nswidth/2)+((k/2)*nspitch), 1, 0, 0,1,nsrough,k+3));
//type should be 1

    /* continue adding ((k/2)*nspitch)*/
    planelist.push_back(newplane(pushind+26+(((k/2)-1)*(14+pushind2))), p5mod14 , nspos-(nspitch -
nswidth/2)-((k/2)*nspitch), nspos-(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimlow, -nsdepth,
0, 0, 1, 0, 0, 0, -1,1,nsrough,-1)); //type should be 1
    planelist.push_back(newplane(pushind+27+(((k/2)-1)*(14+pushind2))), p6mod14 , nspos-(nspitch -
nswidth/2)-((k/2)*nspitch), nspos-(nswidth/2)-((k/2)*nspitch), nsylimhigh, nsylimhigh, -
nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,1,nsrough,-1)); //type should be 1
    planelist.push_back(newplane(pushind+28+(((k/2)-1)*(14+pushind2))), p7mod14 ,
nspos+(nswidth/2)+((k/2)*nspitch), nspos+(nspitch - nswidth/2)+((k/2)*nspitch), nsylimlow,
nsylimlow, -nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,1,nsrough,-1)); //type should be 1
    planelist.push_back(newplane(pushind+29+(((k/2)-1)*(14+pushind2))), p8mod14 ,
nspos+(nswidth/2)+((k/2)*nspitch), nspos+(nspitch - nswidth/2)+((k/2)*nspitch), nsylimhigh,
nsylimhigh, -nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,1,nsrough,-1)); //type should be 1

    planelist.push_back(newplane(pushind+30+(((k/2)-1)*(14+pushind2))), p9mod14 , nspos-
(nswidth/2)-((k/2)*nspitch), nspos+(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimhigh, 0, 0,
0, 0, 1, 0, 0, 0, -1,1,mtrough,k)); // normally mtrough //type should be 1
    planelist.push_back(newplane(pushind+31+(((k/2)-1)*(14+pushind2))), p10mod14 , nspos-
(nswidth/2)+((k/2)*nspitch), nspos+(nswidth/2)+((k/2)*nspitch), nsylimlow, nsylimhigh, 0, 0,
0, 0, 1, 0, 0, 0, -1,1,mtrough,k+1));// normally mtrough //type should be 1

    planelist.push_back(newplane(pushind+32+(((k/2)-1)*(14+pushind2))), p11mod14 , nspos-
(nswidth/2)-((k/2)*nspitch), nspos+(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimhigh, -
nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, 1,5,nsrough,k)); //normally regsys 5
    planelist.push_back(newplane(pushind+33+(((k/2)-1)*(14+pushind2))), p12mod14 , nspos-
(nswidth/2)+((k/2)*nspitch), nspos+(nswidth/2)+((k/2)*nspitch), nsylimlow, nsylimhigh, -
nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, 1,5,nsrough,k+1)); //normally regsys 5

    planelist.push_back(newplane(pushind+34+(((k/2)-1)*(14+pushind2))), p13mod14 , nspos-(nspitch
- nswidth/2)-((k/2)*nspitch), nspos-(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimhigh, -
nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, -1,1,nsgapbotrough,-1)); //type should be 1

```

```

    planelist.push_back(newplane(pushind+35+(((k/2)-1)*(14+pushind2)), p14mod14 ,
    nspos+(nswidth/2)+((k/2)*nspitch), nspos+(nspitch - nswidth/2)+((k/2)*nspitch), nsylimlow,
    nsylimhigh, -nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, -1,1,nsgapbotrough,-1)); //type
    should be 1

    if (ifangrep==1)
    {
        string angrep1="imaginery entrance plane of the nanosheet "+kstr;
        string angrep2="imaginery exist plane of the nanosheet "+kstr;
        string angrep3="imaginery entrance plane of the nanosheet "+kp1str;
        string angrep4="imaginery exist plane of the nanosheet "+kp1str;

        planelist.push_back(newplane(pushind+36+(((k/2)-1)*(14+pushind2)), angrep1 , nspos-
        (nswidth/2)-((k/2)*nspitch), nspos+(nswidth/2)-((k/2)*nspitch), nsylimlow, nsylimlow, -
        nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,4,nsrough,k));
        planelist.push_back(newplane(pushind+37+(((k/2)-1)*(14+pushind2)), angrep2 , nspos-
        (nswidth/2)-((k/2)*nspitch), nspos+(nswidth/2)-((k/2)*nspitch), nsylimhigh, nsylimhigh, -
        nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,4,nsrough,k));

        planelist.push_back(newplane(pushind+38+(((k/2)-1)*(14+pushind2)), angrep3 , nspos-
        (nswidth/2)+((k/2)*nspitch), nspos+(nswidth/2)+((k/2)*nspitch), nsylimlow, nsylimlow, -
        nsdepth, 0, 0, 1, 0, nsylimlow, 0, -1, 0,4,nsrough,k+1));
        planelist.push_back(newplane(pushind+39+(((k/2)-1)*(14+pushind2)), angrep4 , nspos-
        (nswidth/2)+((k/2)*nspitch), nspos+(nswidth/2)+((k/2)*nspitch), nsylimhigh, nsylimhigh, -
        nsdepth, 0, 0, 1, 0, nsylimhigh, 0, 1, 0,4,nsrough,k+1));

    }

}

// define system planes other than nanosheets
if (k==0)
{
    planelist.push_back(newplane(10, "mesa top plane" , -xnegborder, xposborder, -mesawidth/2,
    mesawidth/2, 0, 0, 0, 1, 0, 0, 0, -1,1,mtrough,0)); //normally type 1 //type should be 1
    planelist.push_back(newplane(11, "wafer bottom system border plane" , -2*xnegborder,
    2*xposborder, -2*ynegborder, 2*yposborder, -mesatoptobot, -mesatoptobot, 0, 0, 1, -
    mesatoptobot, 0, 0, 1,fbotype,wbrough,0));
    planelist.push_back(newplane(12, "top plane between the negative y system border and the
    generator" , -xnegborder, xposborder, -ynegborder, -mesabotwidth/2, -mesaheight, -mesaheight,
    0, 0, 1, -mesaheight, 0, 0, -1,1,mtrough,0)); //type should be 1
    planelist.push_back(newplane(13, "top plane between the positive y system border and the
    detector" , -xnegborder, xposborder, mesabotwidth/2, yposborder, -mesaheight, -mesaheight, 0,
    0, 1, -mesaheight, 0, 0, -1,1,mtrough,0)); //type should be 1

}
else
{
    planelist.push_back(newplane(pushind+22+(((k/2)-1)*(14+pushind2)), "top plane between the
    negative x system border and the latest nanosheet gap" , -xnegborder, nspos-(nspitch-
    nswidth/2)-(((k-2)/2)*nspitch), nsylimlow, nsylimhigh, 0, 0, 0, 0, 1, 0, 0, 0, -
    1,1,mtrough,0)); //normally type 1 //type should be 1
    planelist.push_back(newplane(pushind+23+(((k/2)-1)*(14+pushind2)), "top plane between the
    positive x system border and the latest nanosheet gap" , nspos+(nspitch-nswidth/2)+(((k-
    2)/2)*nspitch), xposborder, nsylimlow, nsylimhigh, 0, 0, 0, 0, 1, 0, 0, 0, -1,1,mtrough,0));
    //normally type 1 //type should be 1
    planelist.push_back(newplane(pushind+24+(((k/2)-1)*(14+pushind2)), "top plane between the
    generator and the nanosheets" , -xnegborder, xposborder, -
    ((mesawidth/2)+(0.0001*(mesawidth/2))), nsylimlow, 0, 0, 0, 0, 1, 0, 0, 0, -1,1,mtrough,0));
    //normally type 1 //type should be 1
    planelist.push_back(newplane(pushind+25+(((k/2)-1)*(14+pushind2)), "top plane between the
    detector and the nanosheets" , -xnegborder, xposborder, nsylimhigh, mesawidth/2, 0, 0, 0, 0,
    1, 0, 0, 0, -1,1,mtrough,0)); //normally type 1 //type should be 1
}
}

```

```

    planelist.push_back(newplane(pushind+26+(((k/2)-1)*(14+pushind2)), "wafer bottom system
border plane" , -2*xnegborder, 2*xposborder, -2*ynegborder, 2*yposborder, -mesatoptobot, -
mesatoptobot, 0, 0, 1, -mesatoptobot, 0, 0, 1,fbotype,wbrough,0));
    planelist.push_back(newplane(pushind+27+(((k/2)-1)*(14+pushind2)), "top plane between the
negative y system border and the generator" , -xnegborder, xposborder, -ynegborder, -
mesabotwidth/2, -mesaheight, -mesaheight, 0, 0, 1, -mesaheight, 0, 0, -1,1,mtrough,0));
//normally type 1 //type should be 1
    planelist.push_back(newplane(pushind+28+(((k/2)-1)*(14+pushind2)), "top plane between the
positive y system border and the detector" , -xnegborder, xposborder, mesabotwidth/2,
yposborder, -mesaheight, -mesaheight, 0, 0, 1, -mesaheight, 0, 0, -1,1,mtrough,0)); //normally
type 1 //type should be 1
    if (detratio!=1)
    {
        planelist.push_back(newplane(pushind+29+(((k/2)-1)*(14+pushind2)), "half non-detector plane
at mesa side wall" , detshift-(detwidth/2), detshift+(detwidth/2), (mesawidth/2),
(mesabotwidth/2)-(detratio*((mesabotwidth/2)-(mesawidth/2))), mesaheight*(detratio-1), 0, 0,
mstilt, 1, (mesawidth/2)*mstilt, 0, -mstilt/sqrt(double((mstilt*mstilt)+1)), -
1/sqrt(double((mstilt*mstilt)+1)), 1,msrough,0));
        //planelist.push_back(newplane(pushind+29+(((k/2)-1)*(12+pushind2)), "negative x temporary
barrier plane" , -xnegborder, nspos-(nspitch-nswidth/2)-(((k-2)/2)*nspitch), nsylimlow,
nsylimlow, -mesaheight, 0, 0, 1, 0, nsylimlow, 0, -1, 0,3,nsrough));
        //planelist.push_back(newplane(pushind+30+(((k/2)-1)*(12+pushind2)), "positive x temporary
barrier plane" , nspos+(nspitch-nswidth/2)+(((k-2)/2)*nspitch), xposborder, nsylimlow,
nsylimlow, -mesaheight, 0, 0, 1, 0, nsylimlow, 0, -1, 0,3,nsrough));
    }
    //planelist.push_back(newplane(pushind+28+(((k/2)-1)*(12+pushind2)), "negative x temporary
barrier plane" , -xnegborder, nspos-(nspitch-nswidth/2)-(((k-2)/2)*nspitch), nsylimlow,
nsylimlow, -mesaheight, 0, 0, 1, 0, nsylimlow, 0, -1, 0,3,nsrough));
    //planelist.push_back(newplane(pushind+29+(((k/2)-1)*(12+pushind2)), "positive x temporary
barrier plane" , nspos+(nspitch-nswidth/2)+(((k-2)/2)*nspitch), xposborder, nsylimlow,
nsylimlow, -mesaheight, 0, 0, 1, 0, nsylimlow, 0, -1, 0,3,nsrough));
}
// this function is to generate planes for only a single closed nanosheet
void generatesimplanes(tvector<PLANE> &planelist, double nslength, double nswidth, double
nsdepth, double nsrough, double mtrough, double detdist, double gendist, int botype, int
ifangrep)
{
    double xnegborder, xposborder, ynegborder, yposborder, znegborder, zposborder;
    xnegborder=50000;
    xposborder=50000;
    ynegborder=50000;
    yposborder=50000;
    znegborder=50000;
    zposborder=50000;

    string p0mod5="low x border nanosheet";
    string p1mod5="high x border nanosheet";
    string p2mod5="low y plane of the nanosheet ";
    string p3mod5="top plane of the nanosheet ";
    string p4mod5="high y plane of the nanosheet ";
    string p5mod5="bottom plane of the nanosheet ";

    double mstilt=tan(double(54.75*PI/180));

    planelist.push_back(newplane(0, p0mod5 , -(nswidth/2), -(nswidth/2), -nslength/2, nslength/2,
-nsdepth, 0, 1, 0, 0, -(nswidth/2), 1, 0, 0,1,nsrough,1)); //normally type 1
    planelist.push_back(newplane(1, p1mod5 , (nswidth/2), (nswidth/2), -nslength/2, nslength/2, -
nsdepth, 0, 1, 0, 0, (nswidth/2), -1, 0, 0,1,nsrough,1)); //normally type 1
    planelist.push_back(newplane(2, p4mod5 , -(nswidth/2), (nswidth/2), nslength/2+detdist,
nslength/2+detdist, -nsdepth, 0, 0, 1, 0, (nslength/2)+detdist, 0, -1, 0,2,nsrough,1));

    /* For tilted detector:
    planelist.push_back(newplane(2, p4mod5 , -(nswidth/2), (nswidth/2), nslength/2+detdist,
nslength/2+detdist+(nsdepth/mstilt), -nsdepth, 0, 0, mstilt, 1, ((nslength/2)+detdist)*mstilt,
0, -mstilt/sqrt(double((mstilt*mstilt)+1)), -1/sqrt(double((mstilt*mstilt)+1)), 2,nsrough,1));
    */

```

```

    planelist.push_back(newplane(3, p3mod5 , -(nswidth/2), (nswidth/2), -nslength/2, nslength/2,
    0, 0, 0, 0, 1, 0, 0, 0, -1,1,mtrough,1)); //normally type 1
    planelist.push_back(newplane(4, p2mod5 , -(nswidth/2), (nswidth/2), -nslength/2-gendist, -
    nslength/2-gendist, -nsdepth, 0, 0, 1, 0, -(nslength/2)-gendist, 0, 1, 0,3,nsrough,1)); //--
    normally this way
    // planelist.push_back(newplane(4, p2mod5 , -(500), (500), -nslength/2-gendist, -nslength/2-
    gendist, -nsdepth, 0, 0, 1, 0, -(nslength/2)-gendist, 0, 1, 0,3,nsrough,1)); //-- for limited
    are generator: 1000x1000 nm

    /* For tilted source with 1000 nm wide generator:
    planelist.push_back(newplane(4, p2mod5 , -(500), (500), -nslength/2-gendist-(nsdepth/mstilt),
    -nslength/2-gendist, -nsdepth, 0, 0, -mstilt, 1, ((nslength/2+gendist))*mstilt, 0,
    mstilt/sqrt(double((mstilt*mstilt)+1)), -1/sqrt(double((mstilt*mstilt)+1)),1,nsrough,0));
    */

    // planelist.push_back(newplane(4, p2mod5 , (((-nswidth/2))/4.0)+(((nswidth/2))/4.0),
    (((nswidth/2))/4.0)+(((nswidth/2))/4.0), -nslength/2-gendist, -nslength/2-gendist, -
    nsdepth*double(5.0/8.0), -nsdepth*double(3.0/8.0), 0, 1, 0, -(nslength/2)-gendist, 0, 1,
    0,3,nsrough,1)); //for 16 times reduced area
    // planelist.push_back(newplane(4, p2mod5 , (((-nswidth/2))/4.0)-(((nswidth/2))/4.0),
    (((nswidth/2))/4.0)-(((nswidth/2))/4.0), -nslength/2-gendist, -nslength/2-gendist, -
    nsdepth*double(5.0/8.0), -nsdepth*double(3.0/8.0), 0, 1, 0, -(nslength/2)-gendist, 0, 1,
    0,3,nsrough,1)); //for 16 times reduced area
    // planelist.push_back(newplane(4, p2mod5 , (((-nswidth/2))/4.0), (((nswidth/2))/4.0), -
    nslength/2-gendist, -nslength/2-gendist, -nsdepth*double(5.0/8.0), -nsdepth*double(3.0/8.0),
    0, 1, 0, -(nslength/2)-gendist, 0, 1, 0,3,nsrough,1)); //for 16 times reduced area
    planelist.push_back(newplane(5, p5mod5 , -(nswidth/2), (nswidth/2), -nslength/2, nslength/2, -
    nsdepth, -nsdepth, 0, 0, 1, -nsdepth, 0, 0, 1,botype,nsrough,1));

    planelist.push_back(newplane(6, "positive x system border plane" , xposborder, xposborder, -
    2*ynegborder, 2*yposborder, -2*znegborder, 2*zposborder, 1, 0, 0, xposborder, -1, 0, 0,
    3,0,0));
    planelist.push_back(newplane(7, "negative x system border plane" , -xnegborder, -xnegborder, -
    2*ynegborder, 2*yposborder, -2*znegborder, 2*zposborder, 1, 0, 0, -xnegborder, 1, 0, 0,
    3,0,0));

    planelist.push_back(newplane(8, "positive y system border plane" , -2*xnegborder,
    2*xposborder, yposborder, yposborder, -2*znegborder, 2*zposborder, 0, 1, 0, yposborder, 0, -1,
    0,3,0,0));
    planelist.push_back(newplane(9, "negative y system border plane" , -2*xnegborder,
    2*xposborder, -ynegborder, -ynegborder, -2*znegborder, 2*zposborder, 0, 1, 0, -ynegborder, 0,
    1, 0,3,0,0));

    planelist.push_back(newplane(10, "positive z system border plane" , -2*xnegborder,
    2*xposborder, -2*ynegborder, 2*yposborder, zposborder, zposborder, 0, 0, 1, zposborder, 0, 0,
    -1,3,0,0));
    planelist.push_back(newplane(11, "negative z system border plane" , -2*xnegborder,
    2*xposborder, -2*ynegborder, 2*yposborder, -znegborder, -znegborder, 0, 0, 1, -znegborder, 0,
    0, 1,3,0,0));

    if (ifangrep==1)
    {
        string angrep1="imaginery entrance plane of the nanosheet ";
        string angrep2="imaginery exist plane of the nanosheet ";

        planelist.push_back(newplane(12, angrep1 , -(nswidth/2), (nswidth/2), -nslength/2, -
        nslength/2, -nsdepth, 0, 0, 1, 0, -(nslength/2), 0, -1, 0,4,nsrough,1));
        planelist.push_back(newplane(13, angrep2 , -(nswidth/2), (nswidth/2), nslength/2,
        nslength/2, -nsdepth, 0, 0, 1, 0, (nslength/2), 0, 1, 0,4,nsrough,1));
    }

}

// this is the main function that calls other functions in a rational order

```

```

int main()
{
    tvector<PLANE> planelist(0); // vector variable that stores details for every plane in the
    system. the details are listed in PLANE struct
    tvector<PHANGREP> phforangrep(0); // vector variable that stores details regarding the phonons
    that interact with the nanosheets. See PANGREP struct
    tvector<double> fpath(0); // vector that stores phonon free path values
    tvector<double> freqlist(0); // vector that stores phonon frequency values
    tvector<int> phononsperf_vec(0); // vector that stores number of phonon per frequency or
    frequency distribution
    tvector<double> fypath(0); // vector that stores phonon free path values along y axis only
    tvector<int> fpbin(0); // vector for storing free path distribution
    tvector<double> phonon(8); // vector defining simulated phonon, one phonon exists in the sytem
    at a time
    tvector<double> prephonon(8); // in case failure of the algorithm, the previous phonon
    information to be recalled
    tvector<double> preprephonon(8); // in case double failure of the algorithm, the pre-previous
    phonon information to be recalled
    // For definition of variables below track the on screen input questions
    int recplindmem,recplindmemmem,recplindmemmemmem,
    howmanyns,tzerocnt,recplind,detcount,detcountundns, detcountfrgp,
    syscount,phononstot,phononsperf,freqnum,mxind,mzind,mxsize,mzsize,mdetxind,mdetzind,mdetxsize,
    mdetzsize;
    double starttime, endtime, totaltime,mxstsize,mzstsize,mdetxstsize,mdetzstsize;
    double remtime;
    int i;
    int simorcomp, ifdiff,ifin,botype,fbotype,dethalf,unioirem,plaiden,plaind;
    int ifcosgen, ifcoss Catt,ifpsource,ifrep,iflucky,
    ifangrep,ifangrepdet,ifangrepbin,ifdetrep,ifdetrepbin, ifpffgen, ifpffscatt;
    int bouncein=1;
    double twodelta,binsize, freqsize, specrand, specularity,t,ttemp,vsound, frequency, lambda,
    freqlow,freqhigh,peakfreqems,peakratio, nspos, nslength, nswidth, nspitch, nsdepth, mesawidth,
    mesaheight, nstodet, mesatoptobot, detwidth,detshift, genwidth, xposborder, xnegborder,
    yposborder, ynegborder, nsrough, msrough, mtrough, wbrough,
    nsgapbotrough,gendist,detdist,freepath,freepathy,xold,yold,zold,detratio;
    string fpfilename;
    string angfilename;
    string detfilename, detbinfilename;
    RandGen random;
    // output stream definitions
    ofstream propout;
    ofstream phangrepbinout;
    ofstream phangrepbinin;
    ofstream phangrepout;
    ofstream phangrepouts;
    ofstream phangrepoutsov;
    ofstream phdetrepout;
    ofstream phdetrepbinout;

    // lines below belong to afm surface simulation algorithm.
    ifstream afmr;
    string infnamerough;
    infnamerough="nanosheet surface.txt";
    afmr.open(infnamerough.c_str());
    tmatrix<double> afmrroughness(512,512);
    //cout<< "generated matrix successfully" <<endl;
    //system ("pause");

    int rmatindx, rmatindy;
    rmatindx=0;
    rmatindy=0;

    double prough;
    int rcount;
    rcount = 0;
    while (afmr >> prough)
    {

```

```

//cout<< "got into while loop" <<endl;
//system ("pause");
afmroughness[rmatindx][rmatindy]=prough;
//cout<< "assigned value to matrix" <<endl;
//system ("pause");
rcount++;
rmatindy++;
if (rcount%512 == 0)
{
    //cout << "rmatindy-1= " << rmatindy-1 << " afmroughness[rmatindx][rmatindy]= "
    <<afmroughness[rmatindx][rmatindy-1]<<endl;
    rmatindy=0;
    rmatindx++;
}

}

//cout << "read complete: rcount= " << rcount << " prough= " << prough << "afmroughness512512=
" << afmroughness[511][511] << endl;

// lines above belong to afm surface simulation algorithm.

//system defined output file names
string propfilename= "system_properties.txt";
propout.open(propfilename.c_str(), ios::app);
ofstream phgenrandrep;
string phgenrandrepfn= "phonon_generation_randomness.txt";
phgenrandrep.open(phgenrandrepfn.c_str(), ios::app);
phgenrandrep << "Phonon i vector" <<"\t" <<"Phonon j vector" <<"\t" <<"Phonon k vector" <<"\t"
<<"Phonon x position" <<"\t" <<"Phonon y position" <<"\t" <<"Phonon z position" <<endl;
t=DBL_MAX;
ifstream input;
ifin=0;
string inputfile;
string filename;
string filename2;
// ONE CAN TRACK DEFINITION OF VARIABLE USING THE INPUT QUESTIONS BELOW
cout << "Would you like to input the simulation parameters through a keyboard(0) or a text
file(1)?" << endl;
cin >> ifin;
cout << "Enter name for the input file(enter random string if you don't want to use input
file): "<<endl;
cin >> inputfile;
input.open(inputfile.c_str());
if (ifin==1)
{
    input >> filename;
    input >>filename2;
    input >> ifrep;
    input >>ifpffgen;
    input >>ifpffscatt;
    propout << "Planes Output Filename: " << filename << endl;
    propout << "Transmission Output Filename: " << filename2 << endl;
    propout << "Free path values reported(1) or not(0): " << ifrep << endl;
    propout << "PFF included for phonon generation (1) or not(0): " << ifpffgen << endl;
    propout << "PFF included for phonon scattering (1) or not(0): " << ifpffscatt << endl;
}
else
{
    cout << "Enter name for the output file for the mathematical representation of the phonon
spectrometry system (output file should be in text format, need .txt at the end of filename):
"<<endl;
    cin>> filename;
    cout <<"Enter name for the output file for the transmitted intensity vs frequency data (output
file should be in text format, need .txt at the end of filename): "<<endl;
    cin>>filename2;
}

```



```

cout << "Would you like to report free path values? (1 for yes, 0 for no)" << endl;
cin >> ifrep;
cout << "Would you like to incorporate phonon focusing factors for phonon generation? (1 for
yes, 0 for no)" <<endl;
cin >> ifpffgen;
cout << "Would you like to incorporate phonon focusing factors for phonon scattering? (1 for
yes, 0 for no)" <<endl;
cin >> ifpffscatt;
propout << "Planes Output Filename: " << filename << endl;
propout << "Transmission Output Filename: " << filename2 << endl;
propout << "Free path values reported(1) or not(0): " << ifrep << endl;
propout << "PFF included for phonon generation (1) or not(0): " << ifpffgen << endl;
propout << "PFF included for phonon scattering (1) or not(0): " << ifpffscatt << endl;
}

if (ifrep==1)
{
    if (ifin==1)
    {
        input >> fpfilename;
        input >> binsize;
        propout << "Generic file name for free path files: " << fpfilename << endl;
        propout << "Bin size for free path distribtion: " << binsize << endl;
    }
    else
    {
        cout << "Please enter the file name for free path files (no need for .txt)" << endl;
        cin >> fpfilename;
        cout << "Please enter the bin size for free path distribtion. (a length value like 5 nms)"
<< endl;
        cin>> binsize;
        propout << "Generic file name for free path files: " << fpfilename << endl;
        propout << "Bin size for free path distribtion: " << binsize << endl;
    }
}

if (ifin==1)
{
    input >> simorcomp;
    propout << "System simple(0) or complex(1): " << simorcomp << endl;
    input >> vsound;
    propout << "Speed of sound: " << vsound << "m/s" << endl;
    input >> unioirem;
    propout << "Low to High Uniform Distribution(0) or STJ Emission Spectra Distribution for
Phonon Frequency(1):" <<unioirem <<endl;

    if (unioirem==0)
    {
        input >> freqlow;
        propout << "Start Phonon Frequency: " << freqlow << "Hz" << endl;
        input >> freqhigh;
        propout << "End Phonon Frequency: " << freqhigh << "Hz" << endl;
    }else
    {
        input >> peakfreqems;
        propout << "Peak Phonon Frequency for STJ Emission Spectra: " << peakfreqems << "Hz" <<
endl;
        input >> peakratio;
        propout << "Ratio of peak emission intensity to total intensity: " << peakratio << "Hz" <<
endl;
    }
    input>>freqnum;
    propout << "Number of frequencies to calculate: " << freqnum << endl;
    input >> twodelta;
}

```



```

propout << "2 Delta Value: " << twodelta << "microvolts" << endl;
input>> ifpsource;
propout << "Point Source(1) or Area Source(0): " << ifpsource << endl;
input >> phononstot;
propout << "Total number of phonons to simulate: " << phononstot << endl;
input >> ifcosgen;
propout << "Cosine(1) or Random(0) distribution for generated phonons: " << ifcosgen << endl;
input >> ifcoss Catt;
propout << "Cosine(1) or Random(0) distribution for scattered phonons: " << ifcoss Catt <<
endl;
}
else
{
    cout<< "Would you like to simulate a simple structure with 1 nanosheet or the full system?
(Enter 0 for simple and 1 for full system)" << endl;
    cin >> simorcomp;
    cout << "Please enter speed of sound in transport medium (m/s)" << endl;
    cin >> vsound;
    cout << "Please select: Low to High Uniform Distribution(0) or STJ Emission Spectra
Distribution for Phonon Frequency(1):"<<endl;
    cin >> unioirem;

    if (unioirem==0)
    {
        cout << "Please enter lower frequency limit (in Hz)" << endl;
        cin >> freqlow;
        cout << "Please enter upper frequency limit (in Hz)" << endl;
        cin >> freqhigh;
    }else
    {
        cout << "Please enter peak phonon frequency for STJ emission spectra: " << endl;
        cin >> peakfreqems;
        cout << "Please enter ratio of peak emission intensity to total intensity: " << endl;
        cin >> peakratio;
    }

    cout <<"Please enter the total number of frequencies to calculate" << endl;
    cin>>freqnum;
    cout << "Please enter the value for two delta in microvolts" << endl;
    cin >> twodelta;
    cout << "Would you like point source on the detector plane (Enter 1) or detector plane to be
planar source (Enter 0)?"<< endl;
    cin>> ifpsource;
    cout << "Please enter total number of phonons to be generated" << endl;
    cin >> phononstot;
    cout << "Would you like total random distribution or cosine random distribution for phonon
generation? Enter 0 for totally random and 1 for cosine random" << endl;
    cin >> ifcosgen;
    cout << "Would you like total random distribution or cosine random distribution for phonon
diffusive scattering? Enter 0 for totally random and 1 for cosine random" << endl;
    cin >> ifcoss Catt;
    cout << "PLEASE ENTER ONLY POSITIVE VALUES UNLESS GIVEN OPTION TO ENTER BOTH POSITIVE OR
NEGATIVE VALUES" << endl;
    cout << "PLEASE ALWAYS USE NANOMETERS AS THE UNIT FOR THE DISTANCE INPUTS" << endl;

    propout << "System simple(0) or complex(1): " << simorcomp << endl;
    propout << "Speed of sound: " << vsound << "m/s" << endl;
    propout <<"Low to High Uniform Distribution(0) or STJ Emission Spectra Distribution for
Phonon Frequency(1):" <<unioirem <<endl;
    if (unioirem==0)
    {
        propout << "Start Phonon Frequency: " << freqlow << "Hz" << endl;
        propout << "End Phonon Frequency: " << freqhigh << "Hz" << endl;
    }else
    {

```

```

    propout << "Peak Phonon Frequency for STJ Emission Spectra: " << peakfreqems << "Hz" <<
endl;
    propout << "Ratio of peak emission intensity to total intensity: " << peakratio << "Hz" <<
endl;
}

propout << "Number of frequencies to calculate: " << freqnum << endl;
propout << "2 Delta Value: " << twodelta << "microvolts" << endl;
propout << "Point Source(1) or Area Source(0): " << ifpsource << endl;
propout << "Number of phonons to simulate: " << phononstot << endl;
propout << "Cosine(1) or Random(0) distribution for generated phonons: " << ifcosgen << endl;
propout << "Cosine(1) or Random(0) distribution for scattered phonons: " << ifcoss catt <<
endl;
}
if (simorcomp==1)
{

    if (ifin==1)
    {
        input >> fbotype;
        input >> howmanyins;
        input >> ifangrep;
        propout << "Wafer bottom plane type (Regular plane(1) or System Boundary(3)) : " << fbotype
<<endl;
        propout << "Number of nanosheets: " << howmanyins <<endl;
        propout << "Report nanosheet spesific phonon interaction details(YES=1, NO=0)" <<
ifangrep<<endl;

        if (ifangrep==1)
        {
            input >> angfilename;
            input >> ifangrepdet;
            input >> ifangrepbin;
            propout << "Filename for spesific phonon interaction details:" << angfilename<<endl;
            propout << "Report phonon entrence/exit angles-counts to/from the nanosheet (YES=1, NO=0)"
<< ifangrepdet<<endl;
            propout << "Binned report phonon entrence/exit angles-counts to/from the nanosheet (YES=1,
NO=0)" << ifangrepbin<<endl;

            if (ifangrepbin==1)
            {
                input >> mxsize;
                input >> mzsize;
                propout << "X axis number of steps for binned report phonon entrence/exit angles-counts
to/from the nanosheet: " << mxsize<<endl;
                propout << "Z axis number of steps for binned report phonon entrence/exit angles-counts
to/from the nanosheet: " << mzsize<<endl;
            }
        }

        input >> ifdetrep;
        propout << "Report detector phonon interaction details(YES=1, NO=0)" << ifdetrep<<endl;

        if (ifdetrep==1)
        {
            input >> ifdetrepbin;
            propout << "Binned report for phonon detector interaction details (YES=1, NO=0)" <<
ifdetrepbin<<endl;

            if (ifdetrepbin==1)
            {
                input >> mdetxsize;
                input >> mdetzsize;
                propout << "X axis number of steps for binned report for phonon detector interaction
details: " << mdetxsize<<endl;

```

```

        propout << "Z axis number of steps for binned report for phonon detector interaction
details: " << mdetzsize<<endl;
    }
}

}else
{

    cout << "What is the type of the wafer bottom plane? (Regular plane(1) or System
Boundary(3))" <<endl;
    cin >>fbotype;
    cout << " Please enter the number of nanosheets (only positive odd values or zero) "<< endl;
    cin >> howmanyys;
    cout << "Would you like to report nanosheet spesific phonon interaction details(YES=1, NO=0)"
<< endl;
    cin >> ifangrep;
    cout << "Would you like to report phonon entrance-exit angles to-from nanosheets (YES=1,
NO=0)" << endl;
    cin >> ifangrepdet;

    propout << "Wafer bottom plane type (Regular plane(1) or System Boundary(3)) : " << fbotype
<<endl;
    propout << "Number of nanosheets: " << howmanyys <<endl;

    propout << "Report nanosheet spesific phonon interaction details(YES=1, NO=0)" <<
ifangrep<<endl;

    if (ifangrep==1)
    {

        cout << "Please enter the generic filename for spesific phonon interaction details:" <<
angfilename<<endl;
        cin >> angfilename;
        cout << "Would you like to report phonon entrance-exit angles to-from nanosheets (YES=1,
NO=0)" << endl;
        cin >> ifangrepdet;
        cout << "Would you like the algorithm to prepare binned report phonon entrance-exit angles
to-from nanosheets (YES=1, NO=0)" << endl;
        cin >> ifangrepbin;

        propout << "Filename for spesific phonon interaction details:" << angfilename<<endl;
        propout << "Report phonon entrence/exit angles-counts to/from the nanosheet (YES=1, NO=0)"
<< ifangrepdet<<endl;
        propout << "Binned report phonon entrence/exit angles-counts to/from the nanosheet (YES=1,
NO=0)" << ifangrepbin<<endl;

        if (ifangrepbin==1)
        {
            cout << "Please enter X axis number of steps for binned report for phonon entrence/exit
angles-counts to/from the nanosheet: " << endl;
            cin >> mxsize;
            cout << "Please enter Z axis number of steps for binned report for phonon entrence/exit
angles-counts to/from the nanosheet: " << endl;
            cin >> mzsize;

            propout << "X axis number of steps for binned report for phonon entrence/exit angles-counts
to/from the nanosheet: " << mxsize<<endl;
            propout << "Z axis number of steps for binned report for phonon entrence/exit angles-counts
to/from the nanosheet: " << mzsize<<endl;
        }
    }

    cout << "Would you like to report phonon detector interaction details(YES=1, NO=0)" << endl;
    cin >> ifdetrep;
    propout << "Report detector phonon interaction details(YES=1, NO=0)" << ifdetrep<<endl;

    if (ifdetrep==1)

```

```

{
    cout << "Would you like the algorithm to prepare binned report for phonon detector
interaction details (YES=1, NO=0)" << endl;
    cin >> ifdetrepbin;
    propout << "Binned report for phonon detector interaction details (YES=1, NO=0)" <<
ifdetrepbin<<endl;

    if (ifdetrepbin==1)
    {
        cout << "Please enter X axis number of steps for binned report for phonon detector
interaction: " << endl;
        cin >> mdetxsize;
        cout << "Please enter Z axis number of steps for binned report for phonon detector
interaction: " << endl;
        cin >> mdetzsize;

        propout << "X axis number of steps for binned report for phonon detector interaction
details: " << mdetxsize<<endl;
        propout << "Z axis number of steps for binned report for phonon detector interaction
details: " << mdetzsize<<endl;
    }
}

if (howmanyins != 0)
{
    if (ifin==1)
    {
        input >> nspos;
        input >> nstodet;
        input >> nslength;
        input >> nswidth;
        input >> nspitch;
        input >> nsdepth;
        input >> nsrough;
        input >> nsgapbotrough;

        propout << " The position of the middle nanosheet with respect to the generator: " <<
nspos << endl;
        propout << " The distance between the nanosheets and the generator: "<<nstodet<< endl;
        propout << " The nanosheet length: "<< nslength << endl;
        propout << " The nanosheet width: "<< nswidth << endl;
        propout << " The nanosheet pitch: "<< nspitch << endl;
        propout << " The nanosheet depth: "<< nsdepth<< endl;
        propout << " The nanosheet side wall roughness: "<< nsrough<< endl;
        propout << " The roughness for the bottom of the gap between the nanosheets:
"<<nsgapbotrough<< endl;
    }else
    {
        cout << " Please enter the position of the middle nanosheet with respect to the generator
(if it is shifted right or left with respect to the nanosheet. It can be positive or negative.
Try to make it comparable to the nanosheet pitch" << endl;
        cin >> nspos;
        cout << " Please enter the distance between the nanosheets and the generator "<< endl;
        cin >> nstodet;
        cout << " Please enter the nanosheet length "<< endl;
        cin >> nslength;
        cout << " Please enter the nanosheet width "<< endl;
        cin >> nswidth;
        cout << " Please enter the nanosheet pitch "<< endl;
        cin >> nspitch;
        cout << " Please enter the nanosheet depth "<< endl;
        cin >> nsdepth;
        cout << " Please enter the nanosheet side wall roughness "<< endl;
        cin >> nsrough;
        cout << " Please enter the roughness for the bottom of the gap between the nanosheets "<<
endl;
    }
}

```

```

        cin >> nsgapbotrough;
        propout << " The position of the middle nanosheet with respect to the detector: " << nspos
<< endl;
        propout << " The distance between the nanosheets and the detector: "<<nstodet<< endl;
        propout << " The nanosheet length: "<< nslength << endl;
        propout << " The nanosheet width: "<< nswidth << endl;
        propout << " The nanosheet pitch: "<< nspitch << endl;
        propout << " The nanosheet depth: "<< nsdepth<< endl;
        propout << " The nanosheet side wall roughness: "<< nsrough<< endl;
        propout << " The roughness for the bottom of the gap between the nanosheets:
"<<nsgapbotrough<< endl;
    }
}
else
{
    nspos=0;
    nstodet=100;
    nslength=100;
    nswidth=100;
    nspitch=200;
    nsdepth=200;
    nsrough=5;
    nsgapbotrough=5;
}
if (ifin==1)
{
    input >> mesawidth;
    input >> mesaheight;
    input >> mesatoptobot;
    input >> detwidth;
    input >> detratio;
    input >> detshift;
    input >> genwidth;
    input >> xposborder;
    input >> xnegborder;
    input >> yposborder;
    input >> ynegborder;
    input >> msrough;
    input >> mtrough;
    input >> wbrough;

    propout << "The mesa width "<<mesawidth<< endl;
    propout << "The mesa height "<<mesaheight<< endl;
    propout << "The thickness of The wafer "<<mesatoptobot<< endl;
    propout << "The detector width "<<detwidth<< endl;
    propout << "The ratio of detector coverage on mesa sidewall: "<<detratio<< endl;
    propout << "The detector shift "<<detshift<< endl;
    propout << "The generator width "<<genwidth<< endl;
    propout << "The positive x border for the system: "<<xposborder<< endl;
    propout << "The negative x border for the system: "<<xnegborder<< endl;
    propout << "The positive y border for the system: "<<yposborder<< endl;
    propout << "The negative y border for the system: "<<ynegborder<< endl;
    propout << "The mesa side wall roughness: "<<msrough<< endl;
    propout << "The mesa top wall roughness: "<<mtrough<< endl;
    propout << "The wafer bottom roughness "<<wbrough<< endl;

}
else
{
    cout << " Please enter the mesa width "<< endl;
    cin >> mesawidth;
    cout << " Please enter the mesa height "<< endl;
    cin >> mesaheight;
    cout << " Please enter the thickness of the wafer "<< endl;
    cin >> mesatoptobot;

```

```

    cout << " Please enter the detector width "<< endl;
    cin >> detwidth;
    cout << "What is the ratio of detector coverage on mesa sidewall? (between 0 and 1) "<<
endl;
    cin >> detratio;
    cout << " Please enter the detector shift "<< endl;
    cin >> detshift;
    cout << " Please enter the generator width "<< endl;
    cin >> genwidth;
    cout << " Please enter the positive x border for the system (border to the right of the
detector). Please don't enter something so that your detector or nanosheets do not fit in to
the system "<< endl;
    cin >> xposborder;
    cout << " Please enter the negative x border for the system (border to the left of the
detector). Please don't enter something so that your detector or nanosheets do not fit in to
the system "<< endl;
    cin >> xnegborder;
    cout << " Please enter the positive y border for the system (the one near the detector).
Please don't enter something so that your mesa does not fit in to the system "<< endl;
    cin >> yposborder;
    cout << " Please enter the negative y border for the system (border to the left of the
detector). Please don't enter something so that your mesa does not fit in to the system "<<
endl;
    cin >> ynegborder;
    cout << " Please enter the mesa side wall roughness "<< endl;
    cin >> msrough;
    cout << " Please enter the mesa top wall roughness "<< endl;
    cin >> mtrough;
    cout << " Please enter wafer bottom roughness "<< endl;
    cin >> wbrough;

    propout << "The mesa width "<<mesawidth<< endl;
    propout << "The mesa height "<<mesaheight<< endl;
    propout << "The thickness of The wafer "<<mesatoptobot<< endl;
    propout << "The detector width "<<detwidth<< endl;
    propout << "The ratio of detector coverage on mesa sidewall: "<<detratio<< endl;
    propout << "The detector shift "<<detshift<< endl;
    propout << "The generator width "<<genwidth<< endl;
    propout << "The positive x border for the system: "<<xposborder<< endl;
    propout << "The negative x border for the system: "<<xnegborder<< endl;
    propout << "The positive y border for the system: "<<yposborder<< endl;
    propout << "The negative y border for the system: "<<ynegborder<< endl;
    propout << "The mesa side wall roughness: "<<msrough<< endl;
    propout << "The mesa top wall roughness: "<<mtrough<< endl;
    propout << "The wafer bottom roughness "<<wbrough<< endl;

}
generateplanes(planelist, howmanyns, nspos, nslength, nswidth, nspitch, nsdepth, mesawidth,
mesaheight, nstodet, mesatoptobot, detwidth, genwidth, xposborder, xnegborder, yposborder,
ynegborder, nsrough, msrough, mtrough, wbrough, nsgapbotrough,fbotype,
detshift,detratio,ifangrep);
}
else
{

if (ifin==1)
{
    input >> botype;
    input >> nslength;
    input >> nswidth;
    input >> nsdepth;
    input >> nsrough;

```

```

input >> mtrough;
input >> gendist;
input >> detdist;
input >> ifangrep;

propout << " The type of nanosheet bottom plane(Regular plane(1) or System Boundary(3)): "<<
botype << endl;
propout << " The nanosheet length: "<< nslength << endl;
propout << " The nanosheet width: "<< nswidth << endl;
propout << " The nanosheet depth: "<< nsdepth<< endl;
propout << " The nanosheet side wall roughness: "<< nsrough<< endl;
propout << " The mesa top roughness: "<< mtrough<< endl;
propout << " Generator to nanosheet distance: "<< gendist<< endl;
propout << " Detector to nanosheet distance: "<< detdist<< endl;
propout << "Report nanosheet spesific phonon interaction details(YES=1, NO=0)" <<
ifangrep<<endl;

if (ifangrep==1)
{
input >> angfilename;
input >> ifangrepdet;
input >> ifangrepbin;

propout << "Filename for spesific phonon interaction details:" << angfilename<<endl;
propout << "Report phonon entrence/exit angles-counts to/from the nanosheet (YES=1, NO=0)"
<< ifangrepdet<<endl;
propout << "Binned report phonon entrence/exit angles-counts to/from the nanosheet (YES=1,
NO=0)" << ifangrepbin<<endl;

if (ifangrepbin==1)
{
input >> mxsize;
input >> mzsize;
propout << "X axis number of steps for binned report phonon entrence/exit angles-counts
to/from the nanosheet: " << mxsize<<endl;
propout << "Z axis number of steps for binned report phonon entrence/exit angles-counts
to/from the nanosheet: " << mzsize<<endl;
}
}

input >> ifdetrep;
propout << "Report detector phonon interaction details(YES=1, NO=0)" << ifdetrep<<endl;

if (ifdetrep==1)
{
input >> ifdetrepbin;
propout << "Binned report for phonon detector interaction details (YES=1, NO=0)" <<
ifdetrepbin<<endl;

if (ifdetrepbin==1)
{
input >> mdetxsize;
input >> mdetzsize;
propout << "X axis number of steps for binned report for phonon detector interaction
details: " << mdetxsize<<endl;
propout << "Z axis number of steps for binned report for phonon detector interaction
details: " << mdetzsize<<endl;
}
}

}else
{
cout << "What is the type of the nanosheet bottom plane? (Regular plane(1) or System
boundary(3))" <<endl;

```

```

cin>>botype;
cout <<"For a simple system, dimensions can not be larger than 10000" <<endl;
cout << " Please enter the nanosheet length "<< endl;
cin >> nslength;
cout << " Please enter the nanosheet width "<< endl;
cin >> nswidth;
cout << " Please enter the nanosheet depth "<< endl;
cin >> nsdepth;
cout << " Please enter the nanosheet side wall roughness "<< endl;
cin >> nsrough;
cout << " Please enter the mesa top wall roughness  "<< endl;
cin >> mtrough;
cout <<"Please enter the distance between the nanosheet and the generator" << endl;
cin>>gendist;
cout <<"Please enter the distance between the nanosheet and the detector" << endl;
cin>>detdist;
cout << "Would you like to report nanosheet spesific phonon interaction details(YES=1,
NO=0)" << endl;
cin >> ifangrep;

propout << " The type of nanosheet bottom plane(Regular plane(1) or System Boundary(3)): "<<
botype << endl;
propout << " The nanosheet length: "<< nslength << endl;
propout << " The nanosheet width: "<< nswidth << endl;
propout << " The nanosheet depth: "<< nsdepth<< endl;
propout << " The nanosheet side wall roughness: "<< nsrough<< endl;
propout << " The mesa top roughness: "<< mtrough<< endl;
propout << " Generator to nanosheet distance: "<< gendist<< endl;
propout << " Detector to nanosheet distance: "<< detdist<< endl;
propout << "Report nanosheet spesific phonon interaction details(YES=1, NO=0)" <<
ifangrep<<endl;

if (ifangrep==1)
{
    cout << "Please enter the generic filename for spesific phonon interaction details:" <<
angfilename<<endl;
    cin >> angfilename;
    cout << "Would you like to report phonon enterance-exit angles to-from nanosheets (YES=1,
NO=0)" << endl;
    cin >> ifangrepdet;
    cout << "Would you like the algorithm to prepare binned report phonon enterance-exit angles
to-from nanosheets (YES=1, NO=0)" << endl;
    cin >> ifangrepbin;

    propout << "Filename for spesific phonon interaction details:" << angfilename<<endl;
    propout << "Report phonon entrence/exit angles-counts to/from the nanosheet (YES=1, NO=0)"
<< ifangrepdet<<endl;
    propout << "Binned report phonon entrence/exit angles-counts to/from the nanosheet (YES=1,
NO=0)" << ifangrepbin<<endl;

    if (ifangrepbin==1)
    {
        cout << "Please enter number of X axis steps for binned report for phonon entrence/exit
angles-counts to/from the nanosheet: " << endl;
        cin >> mxsize;
        cout << "Please enter number of Z axis steps for binned report for phonon entrence/exit
angles-counts to/from the nanosheet: " << endl;
        cin >> mzsize;

        propout << "X axis number of steps for binned report for phonon entrence/exit angles-
counts to/from the nanosheet: " << mxsize<<endl;
        propout << "Z axis number of steps for binned report for phonon entrence/exit angles-
counts to/from the nanosheet: " << mzsize<<endl;
    }
}

```



```

    }

    cout << "Would you like to report phonon detector interaction details(YES=1, NO=0)" << endl;
    cin >> ifdetrep;
    propout << "Report detector phonon interaction details(YES=1, NO=0)" << ifdetrep<<endl;

    if (ifdetrep==1)
    {
        cout << "Would you like the algorithm to prepare binned report for phonon detector
interaction details (YES=1, NO=0)" << endl;
        cin >> ifdetrepbin;
        propout << "Binned report for phonon detector interaction details (YES=1, NO=0)" <<
ifdetrepbin<<endl;

        if (ifdetrepbin==1)
        {
            cout << "Please enter X axis number of steps for binned report for phonon detector
interaction: " << endl;
            cin >> mdetxsize;
            cout << "Please enter Z axis number of steps for binned report for phonon detector
interaction: " << endl;
            cin >> mdetzsize;

            propout << "X axis number of steps for binned report for phonon detector interaction
details: " << mdetxsize<<endl;
            propout << "Z axis number of steps for binned report for phonon detector interaction
details: " << mdetzsize<<endl;
        }
    }

    }

    generatesimplanes(planelist, nslength, nswidth, nsdepth, nsrough, mtrough, detdist, gendist, botype, if
angrep);
    howmanyans=1;
}
if (uniorem==0)
{
    unifreqdist(freqlist, phononsperf_vec, phononstot, freqlow, freqhigh, freqnum);
    cout<<"Number of frequencies to be simulated: " << freqlist.size()<<endl;

}
else
{
    freqlow=900000000000; // emission spectra is assumed to start at 90 GHz
    emsfreqdist(freqlist, phononsperf_vec, phononstot, freqlow, peakfreqems, peakratio, freqnum);
    cout<<"Number of frequencies to be simulated: " << freqlist.size()<<endl;
}
vsound=vsound*pow(double(10),double(9));
ofstream planesout;
planesout.open(filename.c_str(), ios::app);
ofstream transmission;
transmission.open(filename2.c_str(), ios::app);
planesout << "Plane Index Number" << "\t" << "Plane Reference Number" << "\t" << "Plane
Discription" << "\t" << "Plane Limits (x)" << "\t" << "Plane Limits (y)" << "\t" <<"Plane
Limits (z)" << "\t" << "Plane equation" << "\t" << "Plane Normal Vector (Shows the direction
towards the transpor medium)" << "\t" << "Wall Type (1-Regular, 2-Detector, 3-System
Boundary)" << "\t" << "Plane Roughness" << endl;
transmission << "Frequency (Hz)" << "\t" << "Voltage (microvolts)" << "\t" << "Escape to
System Border count" << "\t" << "Detected count" << "\t" << "Escape to system border ratio" <<
"\t" << "Detected count ratio"<< endl;

//bin ang report variables start here
int binind, binl, binlrec;

if (ifangrepbin==1)
{

```

```

mxstsize=nswidth/double(mxsize);
mzstsize=nswidth/double(mzsize);
}else
{
mxsize=1;
mzsize=1;
}

tvector<int> bincountin(mxsize*mzsize,0);
tvector<int> bincountout(mxsize*mzsize,0);

int detbinind,detbinl,detbinrec;

if (ifdetrepbin==1)
{
plaiden=-1;

for (plaind=0; plaind < planelist.size(); plaind++)
{
if (planelist[plaind].REGDETSYS==2)
{
plaiden=plaind;
cout << "Detector plane reference number: " << planelist[plaind].REFNUM <<endl;
}
}

if (plaiden==--1)
{
cout << "Detector-phonon interaction report algorithm failed to identify the plane
associated with detector" << endl;
ifdetrepbin=0;
system ("pause");
}else
{
mdetxstsize=(planelist[plaiden].XLIMHIGH-planelist[plaiden].XLIMLOW)/double(mdetxsize);
mdetzstsize=(planelist[plaiden].ZLIMHIGH-planelist[plaiden].ZLIMLOW)/double(mdetzsize);
cout << mdetxstsize << ", " << mdetzstsize <<endl;
}
}else
{
mdetxsize=1;
mdetzsize=1;
}

tvector<int> detbincount(mdetxsize*mdetzsize,0);

//bin ang report variables end here

int plsize=planelist.size();
for (i=0; i < plsize; i++)
{
planesout << i << "\t" << planelist[i].REFNUM << "\t" << planelist[i].PLADISC << "\t" <<
planelist[i].XLIMLOW << " and " << planelist[i].XLIMHIGH << "\t" <<planelist[i].YLIMLOW << "
and " << planelist[i].YLIMHIGH << "\t" << planelist[i].ZLIMLOW << " and " <<
planelist[i].ZLIMHIGH << "\t" <<planelist[i].XPARAM <<"x +" << planelist[i].YPARAM << "y +" <<
planelist[i].ZPARAM << "z = " <<planelist[i].PCONST << "\t" << planelist[i].NIDIRC << "i and
" << planelist[i].NJDIRC << "j and " << planelist[i].NKDIRC << "k" << "\t" <<
planelist[i].REGDETSYS << "\t" << planelist[i].ROUGH << endl;
}
i=0;
t=DBL_MAX;
ttemp=0;

```

```

tzerocnt=0;
int plc;
int swrollback = 0;
int confp=1;
int l,numbins;
int specout=1;
int freqint=0;
int freqintang=0;
double freqindrec=0;
double minfpath;
minfpath=DBL_MAX;
string freqstring,freqfpfn, freqstringang, freqangfn,freqangfnpffin,freqangfnpffout,
freqangfns;
ofstream freepathdist;
iflucky=0;
int lucky;
int leak;
int scatproblem=0;
starttime = time(NULL);
int freqcount=0;
int specind=0;
int speccount=0;
//double erfcal,erfcalran,erfcalf;
int erfind=0;
int phleakcount;
int detcounttot=0;
int syscounttot=0;

//variables related to phonon angle report starts here
//phonon entered ns=1, phonon exited ns=2, phonon entered ns then hit the ns wall=3,
//phonon entered ns then hit the ns exit=4, phonon entered ns then hit ns bottom then hit the
det=5,
//phonon entered then hit ns bottom then the sys boundary=6, phonon entered ns then hit the
ns exit then det=7,
//phonon originated from ns bottom then exited ns then hit the det=8, no history of entering
ns but interacted with ns wall then exited=9,
//history of entering ns and interaction with ns wall then exited=10, phonon that entered
from open bottom and directly hit ns exit=11,
//phonon with ns entrance history originated from ns wall then exited from ns open bottom
then hit sys boundary=12,
//phonon without ns entrance history originated from ns wall then exited from ns open bottom
then hit sys boundary=13,
//phonon originated from ns wall then exited ns then hit detector with ns entrance
history=14,
//phonon originated from ns wall then exited ns then hit detector without ns entrance
history=15, phonon originated from ns bottom then hit ns wall=16,
//phonon originated from ns wall then hit the ns bottom then detector with history of ns
entrance=17,
//phonon originated from ns wall then hit the ns bottom then detector without history of ns
entrance=18
int nsind=1; //ns index that phonon interacted
int phangind=1; //phonon angle report index
int ifnsent=0; //did phonon enter ns at any time
int ifnsext=0; //did phonon exit ns at any time

int countin; //count number of phonons entered to ns
int countout; //count number of phonon exited ns

int nsentnsbotsys=0; //count number of phonons entered ns and exited from bottom to sys
boundary
int nsentnswall=0; //count number of phonon entered ns then hit the ns wall
int nsentnsbotdet=0; //count number of phonon entered ns and hit detector after passing
through ns bottom
int nsentnsexdet=0; //count number of phonon entered ns, exited ns then hit detector
int nsentnsext=0; //count number of phonons entered ns and exited ns
int nsbotnsextdet=0; //count number of phonon entered ns from bottom and exited ns and hit det

```

```

    int nswallnsextnoenthist=0; //count number of phonons hit ns wall and exited ns without ns
    entrance history
    int nswallnsextnenthist=0; //count number of phonons hit ns wall and exited ns with ns
    entrance history
    int nsbotnsextnoenthist=0; //count number of phonons that interacted with ns bottom and exited ns
    int nswallnsbotnsnthist=0; //count number of phonons hit ns wall and escaped from ns bottom
    with ns entrance history
    int nswallnsbotnsnthistnoenthist=0; //count number of phonons hit ns wall and escaped from ns
    bottom without ns entrance history
    int nswallnsextnsdetenthist=0; //count number of phonons hit ns wall and exit ns with ns
    entrance history
    int nswallnsextnsdetnoenthist=0; //count number of phonons hit ns wall and exit ns and hit
    det without ns entrance history
    int nsbotnswall=0; //count number of phonons that interact with ns bottom and hit ns wall
    int nswallnsbotnsdetenthist=0; //count number of phonons hit ns wall and interact with ns
    bottom and hit det with ns entrance history
    int nswallnsbotnsdetnoenthist=0; //count number of phonons hit ns wall and interact with ns
    bottom and hit det without ns entrance history
    int nsentnsextnsdet=0; //count number of phonons entered ns and exited ns and hit mesa top
    and hit det
    int nswallnsextnsdet=0; //count number of phonons hit ns wall and exited ns and hit mesa top
    and hit det
    int nsbotnsextnsdet=0; //count number of phonons interacted ns bottom and exited ns and hit
    mesa top and hit det

//below are vectors associated with variables above

tvector<int> countin_vec(howmanyyns+1);
tvector<int> countout_vec(howmanyyns+1);

tvector<int> nsentnsbotnsnthist_vec(howmanyyns+1);
tvector<int> nsentnswall_vec(howmanyyns+1);
tvector<int> nsentnsbotnsdet_vec(howmanyyns+1);
tvector<int> nsentnsextnsdet_vec(howmanyyns+1);
tvector<int> nsentnsextn_vec(howmanyyns+1);
tvector<int> nsbotnsextnsdet_vec(howmanyyns+1);

tvector<int> nswallnsextnoenthist_vec(howmanyyns+1);
tvector<int> nswallnsextnenthist_vec(howmanyyns+1);
tvector<int> nsbotnsextn_vec(howmanyyns+1);
tvector<int> nswallnsbotnsnthist_vec(howmanyyns+1);
tvector<int> nswallnsbotnsnthistnoenthist_vec(howmanyyns+1);
tvector<int> nswallnsextnsdetenthist_vec(howmanyyns+1);
tvector<int> nswallnsextnsdetnoenthist_vec(howmanyyns+1);
tvector<int> nsbotnswall_vec(howmanyyns+1);
tvector<int> nswallnsbotnsdetenthist_vec(howmanyyns+1);
tvector<int> nswallnsbotnsdetnoenthist_vec(howmanyyns+1);
tvector<int> nsentnsextnsdet_vec(howmanyyns+1);
tvector<int> nswallnsextnsdet_vec(howmanyyns+1);
tvector<int> nsbotnsextnsdet_vec(howmanyyns+1);

tvector<double> phivectotout_vec(howmanyyns+1);
tvector<double> phjvectotout_vec(howmanyyns+1);
tvector<double> phkvectotout_vec(howmanyyns+1);

tvector<double> phivectotin_vec(howmanyyns+1);
tvector<double> phjvectotin_vec(howmanyyns+1);
tvector<double> phkvectotin_vec(howmanyyns+1);

tvector<double> phivectotout_wi_vec(howmanyyns+1);
tvector<double> phjvectotout_wi_vec(howmanyyns+1);
tvector<double> phkvectotout_wi_vec(howmanyyns+1);

tvector<double> phivectotin_wi_vec(howmanyyns+1);
tvector<double> phjvectotin_wi_vec(howmanyyns+1);
tvector<double> phkvectotin_wi_vec(howmanyyns+1);

```

```

for (nsind=0; nsind<=howmany; nsind++)
{

    countin_vec[nsind] =0;
    countout_vec[nsind] =0;

    nsentnsbotsys_vec[nsind] =0;
    nsentnswall_vec[nsind] =0;
    nsentnsbotdet_vec[nsind] =0;
    nsentnsexdet_vec[nsind] =0;
    nsentnsext_vec[nsind] =0;
    nsbotnsextdet_vec[nsind] =0;

    nswallnsext_noenthist_vec[nsind] =0;
    nswallnsext_enthist_vec[nsind] =0;
    nsbotnsext_vec[nsind] =0;
    nswallnsbotsys_enthist_vec[nsind] =0;
    nswallnsbotsys_noenthist_vec[nsind] =0;
    nswallnsextnsdet_enthist_vec[nsind] =0;
    nswallnsextnsdet_noenthist_vec[nsind] =0;
    nsbotnswall_vec[nsind] =0;
    nswallnsbotnsdet_enthist_vec[nsind] =0;
    nswallnsbotnsdet_noenthist_vec[nsind] =0;
    nsentnsextmtnsdet_vec[nsind]=0;
    nswallnsextmtnsdet_vec[nsind]=0;
    nsbotnsextmtnsdet_vec[nsind]=0;

    phivectotout_vec[nsind]=0;
    phjvectotout_vec[nsind]=0;
    phkvectotout_vec[nsind]=0;

    phivectotin_vec[nsind]=0;
    phjvectotin_vec[nsind]=0;
    phkvectotin_vec[nsind]=0;

    phivectotout_wi_vec[nsind]=0;
    phjvectotout_wi_vec[nsind]=0;
    phkvectotout_wi_vec[nsind]=0;

    phivectotin_wi_vec[nsind]=0;
    phjvectotin_wi_vec[nsind]=0;
    phkvectotin_wi_vec[nsind]=0;

}

nsind=1;

double phivectotin, phjvectotin, phkvectotin;

double phivectotin_wi, phjvectotin_wi, phkvectotin_wi;

double phivectotout_wi, phjvectotout_wi, phkvectotout_wi;

double phivectotout, phjvectotout, phkvectotout;

double tempfreq=0;
double tempphrefnum=-1;
double tempfreqex=0;
double tempphrefnumex=-1;
//variables related to phonon angle report ends here

double dotpro;
int freqindex=0;
for (freqindex=0; freqindex < freqlist.size(); freqindex++)

```

```

{
phononsperf=phononsperf_vec[freqindex];
frequency=freqlist[freqindex];
detcountundns=0;
detcountfrgp=0;
detcount=0;
syscount=0;
lucky=0;
leak=0;
lambda=vsound/double(frequency);
lambda= static_cast<double>(lambda*1000000+0.5) / 1000000.0;
cout << lambda<<endl;

// detected phonon detail report
if (ifdetrep==1)
{
    detfilename = filename2+"_"+itoa(int(frequency/100000000.0))+ "GHz_detrep.txt";
    phdetrepout.open(detfilename.c_str(), ios::app);
    phdetrepout << "Phonon Detector Interaction Report at frequency "<< frequency << " Hz" <<
endl;
    phdetrepout << "Phonon X pos"<<"\t" <<"Phonon Y pos"<<"\t" <<"Phonon Z pos"<<"\t" <<"Phonon
i vector"<<"\t" <<"Phonon j vector"<<"\t" <<"Phonon k vector"<< endl;

    if (ifdetrepbin==1)
    {
        detbinfilename = filename2+"_"+itoa(int(frequency/100000000.0))+ "GHz_detbinrep.txt";
        phdetrepbinout.open(detbinfilename.c_str(), ios::app);
        //system ("pause");
    }

}

// detected phonon detail report
for (i=0; i <=phononsperf; i++)
{
    ifnsent=0;
    ifnsext=0;
    generatepls(phonon,planelist,frequency,i,ifpsource,ifcosgen, ifpffgen);
    //generatedefpls(phonon, planelist, frequency, i, 0.4, 1, 0);
    if((freqindex==0)|| (freqindex==(freqlist.size())/2)|| (freqindex==(freqlist.size())))
    {

        if (freqindex != freqindrec)
        {
            phgenrandrep << endl;
        }

        phgenrandrep << phonon[5] <<"\t" <<phonon[6] <<"\t" <<phonon[7] <<"\t" <<phonon[2] <<"\t"
<<phonon[3] <<"\t" <<phonon[4] << endl;

        freqindrec=freqindex;

    }
    recplindmem=4;
    if (ifrep==1)
    {
        freepath=0;
        freepathy=0;
    }

    phleakcount=0;

    while (bouncein==1)
    {
        //intreport="No Intersection";

```

```

//intreportlucky="Lucky!";
for (plc=0; plc < plsize; plc++)
{
    if
    (((plane1list[recplindmem].NSASSOC>0)&&((plane1list[recplindmem].NSASSOC==plane1list[plc].NSASSOC
    )||((plane1list[plc].NSASSOC==0)))||((plane1list[recplindmem].NSASSOC<0)&&(plane1list[plc].NSASSOC
    ==0))||((plane1list[recplindmem].NSASSOC==0))
    {
        if(intersect(ttemp,phonon,plane1list[plc],recplindmem))//,intreport,intreportlucky
        {

            /*cout <<"YES. the bool works" << endl;*/
            if (ttemp==0)
            {
                if ((plane1list[plc].REFNUM != 5) && (plane1list[plc].REFNUM != 6))
                {
                    tzerocnt=tzerocnt+1;
                    //cout <<plane1list[plc].PLADISC<<endl;
                    //cout <<tzerocnt<<endl;
                }
            }
            else if ((t>ttemp)&&(ttemp>0))
            {
                t=ttemp;
                recplind=plc;
            }
        }
    }
    if (tzerocnt==2)
    {
        /*cout << "You are extremely lucky. your phonon was able to hit the intesection line of
        two planes and I don't know what to do :). I will assume this phonon never existed and grant
        you one additional phonon" << endl;
        cout << "Intersection line hitting phonon info: " << "frequency " << phonon[0] << "phonon
        referencenumber " << phonon[1] << "x initial position" << phonon[2] << "y initial position" <<
        phonon[3] << "z initial position" << phonon[4] << "i vector" << phonon[5] << "j vector" <<
        phonon[6] << "k vector" << phonon[7] << ". Interesting plane: "
        <<plane1list[recplind].PLADISC<< endl;
        cout << "The phonon before the intersection line hitting phonon: " << "frequency " <<
        prephonon[0] << "phonon referencenumber " << prephonon[1] << "x initial position" <<
        prephonon[2] << "y initial position" << prephonon[3] << "z initial position" << prephonon[4]
        << "i vector" << prephonon[5] << "j vector" << prephonon[6] << "k vector" << prephonon[7]
        <<endl;
        cout << "The phonon two before the intersection line hitting phonon: " << "frequency " <<
        preprephonon[0] << "phonon referencenumber " << preprephonon[1] << "x initial position" <<
        preprephonon[2] << "y initial position" << preprephonon[3] << "z initial position" <<
        preprephonon[4] << "i vector" << preprephonon[5] << "j vector" << preprephonon[6] << "k
        vector" << preprephonon[7] <<endl;
        //cout << "Diffisive(1) or Reflective (0): " << ifdiff <<endl;
        cout << "Plane that intersection line hitting phonon originated: " <<
        plane1list[recplindmem].PLADISC << endl;*/
        //cout << "Intersection report: "<< endl;
        //cout << intreportlucky << endl;
        iflucky=1;
        bouncein=0;
        i=i-1;
        lucky=lucky+1;
        cout << lucky << " out of " << i<< " phonons hitte the intersection line of two planes"
        <<endl;
        //system ("pause");
    } else if (t==DBL_MAX)
    {
        recplind=0;
        cout<<"*****"<<endl<<endl;
        /*cout << "Leaked phonon info: " << "frequency " << phonon[0] << "phonon referencenumber "
        << phonon[1] << "x initial position" << phonon[2] << "y initial position" << phonon[3] << "z

```

```

initial position" << phonon[4] << "i vector" << phonon[5] << "j vector" << phonon[6] << "k
vector" << phonon[7] <<endl;
    cout << "The phonon before the leaking phonon: " << "frequency " << prephonon[0] <<
"phonon referencenumber " << prephonon[1] << "x initial position" << prephonon[2] << "y
initial position" << prephonon[3] << "z initial position" << prephonon[4] << "i vector" <<
prephonon[5] << "j vector" << prephonon[6] << "k vector" << prephonon[7] <<endl;
    cout << "The phonon two before the leaking phonon: " << "frequency " << preprephonon[0] <<
"phonon referencenumber " << preprephonon[1] << "x initial position" << preprephonon[2] << "y
initial position" << preprephonon[3] << "z initial position" << preprephonon[4] << "i vector"
<< preprephonon[5] << "j vector" << preprephonon[6] << "k vector" << preprephonon[7] <<endl;

    cout << "Diffusive(1) or Reflective (0): " << ifdiff <<endl;
    cout << "Plane that leaked phonon originated: " << planelist[recplindmem].PLADISC <<
endl;*/
    cout << "there is a leak in your system. the simulation is not reliable" << endl;
    //cout << "Intersection report: "<< endl;
    //cout << intreport << endl;
    /*cout << "Roll back to the last good phonon"<<endl;*/
    cout<<"*****"<<endl<<endl;
    phonon=preprephonon;
    swrollback=1;
    leak=leak+1;
    cout << leak << " out of " << i<< " phonons leaked from the system" <<endl;
    if (phleakcount==2)
    {
        iflucky=1;
        bouncein=0;
        i=i-1;
    }
    phleakcount=phleakcount+1;
    iflucky=1;
    bouncein=0;
    i=i-1;
    /*system ("pause");*/
} else
{
    if ((ifrep==1)&&(confp==1))
    {
        xold=phonon[2];
        yold=phonon[3];
        zold=phonon[4];
    }

    phonon[2]=(t*phonon[5]) + phonon[2];
    phonon[3]=(t*phonon[6]) + phonon[3];
    phonon[4]=(t*phonon[7]) + phonon[4];

    if(t==0)
    {
        cout << "t value is 0"<<endl;
        cout << "current plane: " << planelist[recplind].PLADISC << ". previous plane: " <<
planelist[recplindmem].PLADISC << endl;
    }

    if ((ifrep==1)&&(confp==1))
    {
        freepath=freepath+ sqrt(double(((phonon[2]-xold)*(phonon[2]-xold))+((phonon[3]-
yold)*(phonon[3]-yold))+((phonon[4]-zold)*(phonon[4]-zold))));
        freepathy=freepathy+(yold-phonon[3]);
    }

    dotpro=(planelist[recplind].NIDIRC*phonon[5])+(planelist[recplind].NJDIRC*phonon[6])+(planelis
t[recplind].NKDIRC*phonon[7]);

    if (dotpro > 0)
    {

```



```

    if ((planelist[recplind].REGDETSYS != 4)&&(planelist[recplind].REGDETSYS != 5))
    {
        cout << "Phonon arriving plane -" << planelist[recplind].PLADISC << "- from back because
dot product is " << dotpro << endl;

        cout<<"*****"<<endl<<endl;
        cout << "Backhitting phonon info: " << "frequency " << phonon[0] << "phonon
referencenumber " << phonon[1] << "x initial position" << phonon[2] << "y initial position" <<
phonon[3] << "z initial position" << phonon[4] << "i vector" << phonon[5] << "j vector" <<
phonon[6] << "k vector" << phonon[7] << endl;
        cout << "The phonon before the Backhitting phonon: " << "frequency " << prephonon[0] <<
"phonon referencenumber " << prephonon[1] << "x initial position" << prephonon[2] << "y
initial position" << prephonon[3] << "z initial position" << prephonon[4] << "i vector" <<
prephonon[5] << "j vector" << prephonon[6] << "k vector" << prephonon[7] << endl;
        cout << "The phonon two before the Backhitting phonon: " << "frequency " <<
preprephonon[0] << "phonon referencenumber " << preprephonon[1] << "x initial position" <<
preprephonon[2] << "y initial position" << preprephonon[3] << "z initial position" <<
preprephonon[4] << "i vector" << preprephonon[5] << "j vector" << preprephonon[6] << "k
vector" << preprephonon[7] << endl;
        cout << "Diffusive(1) or Reflective (0): " << ifdiff << endl;
        cout << "Plane that Backhitting phonon originated: " << planelist[recplindmem].PLADISC <<
endl;
        cout << "there could be leak in your system. the simulation is not reliable" << endl;
        cout<<"*****"<<endl<<endl;

    }
}

//if condition below checks if the intersecting wall is regular system wall
if (planelist[recplind].REGDETSYS==1)
{
    if (ifangrep==1)
    {
        if (planelist[recplind].NSASSOC>0)
        {
            if ((planelist[recplindmem].REGDETSYS==4)&&(planelist[recplindmem].NJDIRC == -1))
            {
                phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 3, phonon[2],
phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
            }

            if (planelist[recplindmem].REGDETSYS==5)
            {
                phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 3, phonon[2],
phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
            }
        }
    }
    speccount=speccount+1;
    //erfcalf=gsl_sf_erf(0.125*lambda/(planelist[recplind].ROUGH*(pow(Pi,1.5))));
    //erfcalf=erf((0.125*lambda/(planelist[recplind].ROUGH*(pow(Pi,1.5)))),100000);
    //erfcalf=ferf(0.125*lambda/(planelist[recplind].ROUGH*(pow(Pi,1.5))));
    //define specularity below:
    if (planelist[recplind].ROUGH!=0)
    {

        //specularity=0.221556731*lambda*exp(0.015625*(lambda*lambda)/((planelist[recplind].ROUGH*plan
elist[recplind].ROUGH)*(PI*PI*PI)))*(1-
ferf(0.125*lambda/(planelist[recplind].ROUGH*(pow(Pi,1.5)))/((planelist[recplind].ROUGH*(pow(
PI,1.5)))); // Poisson's roughness dist

        //specularity=1/sqrt(double(1+double((pow((4.00*PI*planelist[recplind].ROUGH/lambda),2)))));
        // Gaussian roughness dist

```

```

        specularity=exp(-
double(16.0)*(PI*PI)*(planelist[recplind].ROUGH*planelist[recplind].ROUGH)/(lambda*lambda));
// delta function dist
    /*if (frequency>300000000000)
    {

    //specularity=1/sqrt(double(1+double(PI*(pow((4.00*PI*planelist[recplind].ROUGH/lambda),2))))))
; // Gaussian roughness dist
        specularity=0.7282; //For STJ emission spectra with peaks freq at 400 GHz for 1 nm
roughness
    }else
    {
        /*
        //specularity=0.5;
        //}
        //specularity=0;
    }
else
{
    specularity=1;
}

if (specout==1)
{
    cout <<"Phonon Frequency: "<<frequency<< " . Specularity: " << specularity<<endl;
    specout=0;
}
//check if calculated specularity is indefinite and record indefinite number issues
if (specularity!=specularity)
{

    specind=specind+1;
}

specrand=random.RandReal(0,1);
if(dotpro>0)
{
    cout << "Phonon arriving plane -" << planelist[recplind].PLADISC << "- from back because
dot product is " <<dotpro <<endl;
    syscount=syscount+1;
    bouncein=0;
}else
{
    //if condition below determine if the phonon surface interactions will be diffusive or
reflective
    if (specrand <=specularity)
    {
        specrefl(phonon, planelist[recplind],scatproblem); //covert phonon to specularly
reflected phonon
        ifdiff=0;
        if (scatproblem==1)
        {
            bouncein=0;
            i=i-1;
        }
    } else
    {
        diffscatt(phonon,planelist[recplind],ifcoss Catt,scatproblem,ifpffscatt); //normall
diffscat //covert phonon to diffusively scattered phonon
        //surfsimcone(phonon, planelist[recplind]);
        //surfsimafm(phonon, planelist[recplind], afmroughness, 1, 1);
        ifdiff=1;
        confp=0;
        if (scatproblem==1)
        {
            bouncein=0;
            i=i-1;
        }
    }
}

```

```

    }
  }
}
//if condition below checks if the intersecting wall is detector wall
else if (planelist[recplind].REGDETSYS==2)
{
  if (ifangrep==1)
  {
    if (planelist[recplindmem].REGDETSYS==5)
    {
      if ((planelist[recplindmemmem].REGDETSYS==4)&&(planelist[recplindmemmem].NJDIRC == -1))
      {
        phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 5, phonon[2],
        phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
      }else if
      ((planelist[recplindmemmem].REGDETSYS==1)&&(planelist[recplindmemmem].NSASSOC>0))
      {
        if (ifnsent==0)
        {
          phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 17,
          phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
        }else
        {
          phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 18,
          phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
        }
      }
    }
  }

  if ((planelist[recplindmem].REGDETSYS==4)&&(planelist[recplindmem].NJDIRC == 1))
  {
    if ((planelist[recplindmemmem].REGDETSYS==4)&&(planelist[recplindmemmem].NJDIRC == -1))
    {
      phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 7, phonon[2],
      phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
    } else if
    ((planelist[recplindmemmem].REGDETSYS==1)&&(planelist[recplindmemmem].NSASSOC>0))
    {
      if (ifnsent==1)
      {
        phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 14,
        phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
      }else
      {
        phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 15,
        phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
      }
    }
  } else if (planelist[recplindmemmem].REGDETSYS==5)
  {
    phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 8, phonon[2],
    phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
  }
}

}

if ((planelist[recplindmem].NSASSOC==0)&&(planelist[recplindmem].REGDETSYS==1))
{
  if ((planelist[recplindmemmem].REGDETSYS==4)&&(planelist[recplindmemmem].NJDIRC == 1))
  {

```

```

    if ((planelist[recplindmemmem].REGDETSYS==4)&&(planelist[recplindmemmem].NJDIRC
== -1))
    {
        phforangrep.push_back(newphforangrep(planelist[recplindmemmem].NSASSOC, 19,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
    }
    else if (planelist[recplindmem].REGDETSYS==1)
    {
        phforangrep.push_back(newphforangrep(planelist[recplindmemmem].NSASSOC, 20,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
    }
    else if (planelist[recplindmem].REGDETSYS==5)
    {
        phforangrep.push_back(newphforangrep(planelist[recplindmemmem].NSASSOC, 21,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
    }
    } else if (planelist[recplindmem].REFNUM==4)
    {

        detcountundns=detcountundns+1;

    }

}

if (planelist[recplindmem].NSASSOC== -1)
{

    detcountfrgp=detcountfrgp+1;

}

}

if (ifdetrep==1)
{
    phdetrepout << phonon[2] <<"\t" << phonon[3] <<"\t" << phonon[4] <<"\t" << phonon[5]
<<"\t" << phonon[6] <<"\t" << phonon[7] << endl;
    if(ifdetrepbin==1)
    {
        // binned output for detrep algorithm
        mdetxind=int(phonon[2]-planelist[recplind].XLIMLOW)/mdetxstsize;
        mdetzind=int(-phonon[4])/mdetzstsize;
        //system ("pause");
        detbincount[mdetxind+(mdetzind*mdetxsize)]=detbincount[mdetxind+(mdetzind*mdetxsize)]+1;
        //system ("pause");
        // binned output for detrep algorithm
        //cout << detbincount.size() <<endl;
    }
}

//if((int(100000.0*phonon[2]) == -92615364) && (int(100000.0*phonon[3]) == 378269205) &&
(int(100000.0*phonon[4]) == -40000000))

//if((int(100000.0*phonon[2]) == 242615364) && (int(100000.0*phonon[3]) == 378269205) &&
(int(100000.0*phonon[4]) == -40000000))
//if((int(100000.0*phonon[2]) == -32615364) && (int(100000.0*phonon[3]) == 378269205) &&
(int(100000.0*phonon[4]) == -40000000))
//{
//cout << "counted as detected" <<endl;

```

```

    detcount=detcount+1;
    //}

    /*cout<<"Detected!! Number of total detected so far: " << detcount<< " Ratio of detected
to total so far " << double(detcount/(detcount+syscount))<<endl;
    system ("pause");*/
    bouncein=0;
    //if condition below checks if the intersecting wall is boundary system wall
    } else if (planelist[recplind].REGDETSYS==3)
    {
        if (ifangrep==1)
        {
            if (planelist[recplindmem].REGDETSYS==5)
            {
                if ((planelist[recplindmemmem].REGDETSYS==4)&&(planelist[recplindmemmem].NJDIRC == -1))
                {
                    phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 6, phonon[2],
phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
                } else
                {
                    if (ifnsent==1)
                    {
                        phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 12,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
                    } else
                    {
                        phforangrep.push_back(newphforangrep(planelist[recplindmem].NSASSOC, 13,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));
                    }
                }
            }
        }
    }
    syscount=syscount+1;
    /*cout<<"Escaped!! Number of total escaped so far: " << syscount<< " Ratio of escaped to
total so far " << double(syscount/(detcount+syscount))<<endl;
    system ("pause");*/
    bouncein=0;
    } else if (planelist[recplind].REGDETSYS==4)
    {
        if (planelist[recplind].NJDIRC == -1)
        {
            if (phonon[6] >= 0)
            {
                phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 1, phonon[2],
phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));

                if ((tempfreq==phonon[0])&&(tempphrefnum==phonon[1]))
                {
                    cout << "The same phonon re-entered a nanosheet" << endl;
                    cout << "Re-entering phonon info: " << "frequency " << prephonon[0] << "phonon
referencenumber " << prephonon[1] << "x initial position" << prephonon[2] << "y initial
position" << prephonon[3] << "z initial position" << prephonon[4] << "i vector" <<
prephonon[5] << "j vector" << prephonon[6] << "k vector" << prephonon[7] <<endl;
                }

                tempfreq=phonon[0];
                tempphrefnum=phonon[1];
                ifnsent=1;
            }
        }
    }

```

```

    }
    else
    {
        if (phonon[6] >= 0)
        {
            if ((tempfreqex==phonon[0])&&(tempphrefnumex==phonon[1]))
            {
                cout << "The same phonon re-exited a nanosheet" << endl;
                cout << "Re-exiting phonon info: " << "frequency " << prephonon[0] << "phonon
referencenumber " << prephonon[1] << "x initial position" << prephonon[2] << "y initial
position" << prephonon[3] << "z initial position" << prephonon[4] << "i vector" <<
prephonon[5] << "j vector" << prephonon[6] << "k vector" << prephonon[7] <<endl;
            }

            tempfreqex=phonon[0];
            tempphrefnumex=phonon[1];

            phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 2, phonon[2],
phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));

            ifnsext=1;

            if ((planelist[recplindmem].REGDETSYS==4)&&(planelist[recplindmem].NJDIRC == -1))
            {

                phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 4,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));

                if(planelist[recplindmem].NSASSOC != planelist[recplind].NSASSOC)
                {
                    cout << "There is something wrong with angrep algorithm, phonon
entering from one sheet can not directly exit from another one" <<endl;

                }

            } else
            {

                if(planelist[recplindmem].REGDETSYS==1)
                {

                    if(ifnsext==1)
                    {

                        phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 10, phonon[2], phonon[3],
phonon[4], phonon[5], phonon[6], phonon[7]));
                    }else
                    {

                        phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 9, phonon[2], phonon[3],
phonon[4], phonon[5], phonon[6], phonon[7]));
                    }
                }else
                {
                    if(planelist[recplindmem].REGDETSYS != 5)
                    {
                        cout << "There is something wrong with angrep algorithm, if
phonon is not coming from enterence or walls of ns it can only come from bottom of ns" <<endl;

                    }

                    phforangrep.push_back(newphforangrep(planelist[recplind].NSASSOC, 11,
phonon[2], phonon[3], phonon[4], phonon[5], phonon[6], phonon[7]));

                }

            }
        }
    }
}

```

```

    }
}

//cout << "*****" << crntplane.PLADISC << "?????????"<<endl;
//cout << "*****" << crntplane.NSASSOC << "?????????"<<endl;

}

else
{
    if (planelist[recplind].REGDETSYS!=5)
    {
        cout << "REGDETSYS values can't be anything but 1,2,3,4 or 5. Check your planes." <<
endl;
    }
}

recplindmemmemmem=recplindmemmem;
recplindmemmem=recplindmem;
recplindmem=recplind;

}

if (swrollback==1)
{
    recplindmem=recplindmemmem;
}

swrollback=0;
preprephonon=prephonon;
prephonon=phonon;
t=DBL_MAX;
ttemp=0;
tzerocnt=0;

}

if ((ifrep==1) && ((iflucky!=1)|| (scatproblem==1)))
{
    fpath.push_back(freepath);
    fypath.push_back(freepathy);
}
iflucky=0;
scatproblem=0;
confp=1;
bouncein=1;
if ((i%(int)(phononsperf/10)))==0)
{
    cout << specind << "out of " << speccount << " specularly calculations were indefinete
which makes %" << (double(specind)/double(i))*100 <<endl;
    //cout << erfind << "out of " << speccount << " error function calculations were indefinete
which makes %" << (double(erfind)/double(i))*100 <<endl;
    cout << "%" << (double(i)/double(phononsperf))*100 << " completed for this frequency." <<
endl;
}

```

```

    cout <<"%" <<
double((double(syscount+detcount+syscounttot+detcounttot)/double(phononstot))*100)<< " completed
in total" << endl;
    endtime = time(NULL);
    totaltime = endtime - starttime;
    cout << "Elapsed Time: " << int(totaltime)/60<<" minutes, "<< int(totaltime)%60 << "
seconds." << endl;

    remtime=((double(totaltime)/(double(syscount+detcount+syscounttot+detcounttot)/double(phononstot
))))-totaltime;
    cout << "Remaining Time: " << int(remtime)/60<<" minutes, "<< int(remtime)%60 << "
seconds." << endl;
    cout << "Size of phforangrep vector: " <<phforangrep.size() << ". Capacity of phforangrep
vector: " << phforangrep.capacity() << endl;
}

}
speccount=0;
//erfind=0;
specind=0;
specout=1;

// Free path distribution report algorithm starts here
if (ifrep==1)
{
    fpbin.clear();
    fpbin=reportfpath(fpath,phononsperf,binsize,frequency,minfpath,numbins);
    freqint=freqint+1;
    freqstring=itoa(freqint);
    freqfpfn = fpfilename+freqstring+".txt";
    freepathdist.open(freqfpfn.c_str(), ios::app);
    freepathdist << "Free path distribution for frequency "<< frequency << " Hz" << endl;
    freepathdist << "Bin Number" << "\t" << "Start Free Path"<<"\t" <<"End Free Path"<<"\t" <<"#
phonons in the bin"<<"\t" <<"# phonons in the bin+1"<<endl;

    for (l=0;l<numbins;l++)
    {
        freepathdist << l << "\t" << (l*binsize)+minfpath << "\t" << ((l+1)*binsize)+minfpath <<
"\t" << fpbin[l]<< "\t" << (fpbin[l]+1) <<endl;
    }

    freepathdist.close();

    fpbin.clear();
    fpbin=reportfpath(fypath,phononsperf,binsize,frequency,minfpath,numbins);
    freqfpfn = fpfilename+freqstring+"Y.txt";
    freepathdist.open(freqfpfn.c_str(), ios::app);
    freepathdist << "Free path distribution for frequency "<< frequency << " Hz" << endl;
    freepathdist << "Bin Number" << "\t" << "Start Free Path"<<"\t" <<"End Free Path"<<"\t" <<"#
phonons in the bin"<<"\t" <<"# phonons in the bin+1"<<endl;
    for (l=0;l<numbins;l++)
    {
        freepathdist << l << "\t" << (l*binsize)+minfpath << "\t" << ((l+1)*binsize)+minfpath <<
"\t" << fpbin[l]<< "\t" << (fpbin[l]+1) <<endl;
    }

    freepathdist.close();
    fypath.clear();
    fpath.clear();
}
if (ifangrep==1)
{
    freqintang=freqintang+1;
    freqstringang=itoa(freqintang);
    freqangfn = angfilename+freqstringang+".txt";
    freqangfnpffin = angfilename+freqstringang+"pff_in.txt";
    freqangfnpffout = angfilename+freqstringang+"pff_out.txt";
}

```





```

for (plaind=0; plaind < planelist.size(); plaind++)
{
    if (planelist[plaind].NSASSOC==nsind)
    {
        //plaiden=plaind;

        if
((planelist[plaind].NIDIRC==0)&&(planelist[plaind].NJDIRC==1)&&(planelist[plaind].NKDIRC==0)&&
(planelist[plaind].REGDETSYS==4))
        {
            plaiden=plaind;
        }

    }

}

if (plaiden==--1)
{
    cout << "Angle report algorithm failed to identify the plane associated with ns index" <<
endl;
    system ("pause");
}

phivectotin=0;
phjvectotin=0;
phkvectotin=0;
countin=0;

phivectotout=0;
phjvectotout=0;
phkvectotout=0;

phivectotin_wi=0;
phjvectotin_wi=0;
phkvectotin_wi=0;

phivectotout_wi=0;
phjvectotout_wi=0;
phkvectotout_wi=0;

countout=0;
if (ifangrepdet==1)
{
    phangrepout << "Phonon Angle and Count Report for the enterence of nanosheet "<< nsind <<
" at frequency "<< frequency << " Hz" << endl;
    if (simorcomp==0)
    {
        phangrepout << "Phonon X pos"<<"\t" <<"Phonon Y pos"<<"\t" <<"Phonon Z pos"<<"\t"
<<"Phonon i vector"<<"\t" <<"Phonon j vector"<<"\t" <<"Phonon k vector"<<"\t" <<"Angle between
i vector and y axis"<<"\t" <<"Angle between k vector and y axis"<<"\t" <<"Angle between x=0-
xpos and y axis"<<"\t" <<"Angle between z=0-zpos and y axis"<< endl;
    }else
    {
        phangrepout << "Phonon X pos"<<"\t" <<"Phonon Y pos"<<"\t" <<"Phonon Z pos"<<"\t"
<<"Phonon i vector"<<"\t" <<"Phonon j vector"<<"\t" <<"Phonon k vector"<< endl;
    }
}

for (phangind=0; phangind < phforangrep.size() ; phangind++)
{
    //cout << "*****" << phforangrep[phangind].XPOS <<"*****"<<endl;

```

```

if (phforangrep[phangind].INCNSTYPE==1)
{
  if (phforangrep[phangind].INCNSIND==nsind)
  {
    if (ifangrepdet==1)
    {
      if (simorcomp==0)
      {
        phangrepout << phforangrep[phangind].XPOS <<"\t" << phforangrep[phangind].YPOS<<"\t"
        << phforangrep[phangind].ZPOS<<"\t" << phforangrep[phangind].IVEC <<"\t" <<
        phforangrep[phangind].JVEC<<"\t" << phforangrep[phangind].KVEC <<"\t" <<
        180.0*asin(phforangrep[phangind].IVEC/sqrt((phforangrep[phangind].IVEC*phforangrep[phangind].I
        VEC)+(phforangrep[phangind].JVEC*phforangrep[phangind].JVEC))/PI <<"\t" <<
        180.0*asin(phforangrep[phangind].KVEC/sqrt((phforangrep[phangind].KVEC*phforangrep[phangind].K
        VEC)+(phforangrep[phangind].JVEC*phforangrep[phangind].JVEC))/PI <<"\t" <<
        180.0*atan(phforangrep[phangind].XPOS/(phforangrep[phangind].YPOS+(nslength/2.0)+gendist))/PI
        <<"\t" <<
        180.0*atan((phforangrep[phangind].ZPOS+(nsdepth/2.0))/(phforangrep[phangind].YPOS+(nslength/2.
        0)+gendist))/PI << endl;
      }else
      {
        phangrepout << phforangrep[phangind].XPOS <<"\t" << phforangrep[phangind].YPOS<<"\t"
        << phforangrep[phangind].ZPOS<<"\t" << phforangrep[phangind].IVEC <<"\t" <<
        phforangrep[phangind].JVEC<<"\t" << phforangrep[phangind].KVEC << endl;
      }
      if (ifangrepbin==1)
      {
        // binned output for angrep algorithm
        mxind=int((phforangrep[phangind].XPOS-planelist[plaiden].XLIMLOW)/mxstsize);
        mzind=int((-phforangrep[phangind].ZPOS)/mzstsize);
        bincountin[mxind+(mzind*mysize)]=bincountin[mxind+(mzind*mysize)]+1;
        // binned output for angrep algorithm
      }
    }

    phivectotin=phivectotin+abs(phforangrep[phangind].IVEC);
    phjvectotin=phjvectotin+phforangrep[phangind].JVEC;
    phkvectotin=phkvectotin+phforangrep[phangind].KVEC;
    countin++;
  }
}

}

if (ifangrepdet==1)
{
  phangrepout << endl;
  phangrepout << "Phonon Angle and Count Report for the exit of nanosheet "<< nsind << " at
  frequency "<< frequency << " Hz" << endl;
  if (simorcomp==0)
  {
    phangrepout << "Phonon X pos"<<"\t" <<"Phonon Y pos"<<"\t" <<"Phonon Z pos"<<"\t"
    <<"Phonon i vector"<<"\t" <<"Phonon j vector"<<"\t" <<"Phonon k vector"<<"\t" <<"Angle between
    i vector and y axis"<<"\t" <<"Angle between k vector and y axis"<<"\t" <<"Angle between x=0-
    xpos and y axis"<<"\t" <<"Angle between z=0-zpos and y axis"<< endl;
  }else
  {
    phangrepout << "Phonon X pos"<<"\t" <<"Phonon Y pos"<<"\t" <<"Phonon Z pos"<<"\t"
    <<"Phonon i vector"<<"\t" <<"Phonon j vector"<<"\t" <<"Phonon k vector"<< endl;
  }
}
for (phangind=0; phangind < phforangrep.size() ; phangind++)
{
  if (phforangrep[phangind].INCNSTYPE==2)
  {
    if (phforangrep[phangind].INCNSIND==nsind)
    {
      if (ifangrepdet==1)

```

```

    {
        if (simorcomp==0)
        {
            phangrepout << phforangrep[phangind].XPOS <<"\t" << phforangrep[phangind].YPOS<<"\t"
            << phforangrep[phangind].ZPOS<<"\t" << phforangrep[phangind].IVEC <<"\t" <<
            phforangrep[phangind].JVEC<<"\t" << phforangrep[phangind].KVEC <<"\t" <<
            180.0*asin(phforangrep[phangind].IVEC/sqrt((phforangrep[phangind].IVEC*phforangrep[phangind].I
            VEC)+(phforangrep[phangind].JVEC*phforangrep[phangind].JVEC))/PI <<"\t" <<
            180.0*asin(phforangrep[phangind].KVEC/sqrt((phforangrep[phangind].KVEC*phforangrep[phangind].K
            VEC)+(phforangrep[phangind].JVEC*phforangrep[phangind].JVEC))/PI <<"\t" <<
            180.0*atan(phforangrep[phangind].XPOS/(phforangrep[phangind].YPOS+(nslength/2.0)+gendist))/PI
            <<"\t" <<
            180.0*atan((phforangrep[phangind].ZPOS+(nsdepth/2.0))/(phforangrep[phangind].YPOS+(nslength/2.
            0)+gendist))/PI << endl;
        }
        else
        {
            phangrepout << phforangrep[phangind].XPOS <<"\t" << phforangrep[phangind].YPOS<<"\t"
            << phforangrep[phangind].ZPOS<<"\t" << phforangrep[phangind].IVEC <<"\t" <<
            phforangrep[phangind].JVEC<<"\t" << phforangrep[phangind].KVEC << endl;
        }

        // binned output for angrep algorithm
        if (ifangrepbin==1)
        {
            mxind=int((phforangrep[phangind].XPOS-planelist[plaiden].XLIMLOW)/mxstsize);
            mzind=int((-phforangrep[phangind].ZPOS)/mzstsize);
            bincountout[mxind+(mzind*mxsize)]=bincountout[mxind+(mzind*mxsize)]+1;
        }
        // binned output for angrep algorithm
    }
    phivectotout=phivectotout+phforangrep[phangind].IVEC;
    phjvectotout=phjvectotout+phforangrep[phangind].JVEC;
    phkvectotout=phkvectotout+phforangrep[phangind].KVEC;
    countout++;
}
}
if (ifangrepdet==1)
{
    phangrepout << endl;
}

for (phangind=0; phangind < phforangrep.size() ; phangind++) //phonon entered ns=1, phonon
exited ns=2, phonon entered ns then hit the ns wall=3, phonon entered ns then hit the ns
exit=4, phonon entered ns then hit ns bottom then hit the det=5, phonon entered then hit ns
bottom then the sys boundary=6, phonon entered ns then hit the ns exit then det=7, phonon
originated from ns bottom then exited ns then hit the det=8, no history of entering ns but
interacted with ns wall then exited=9, history of entering ns and interaction with ns wall
then exited=10, phonon that entered from open bottom and directly hit ns exit=11, phonon with
ns entrance history originated from ns wall then exited from ns open bottom then hit sys
boundary=12, phonon without ns entrance history originated from ns wall then exited from ns
open bottom then hit sys boundary=13, phonon originated from ns wall then exited ns then hit
detector with ns entrance history=14, phonon originated from ns wall then exited ns then hit
detector without ns entrance history=15, phonon originated from ns bottom then hit ns
wall=16, phonon originated from ns wall then hit the ns bottom then detector with history of
ns entrance=17, phonon originated from ns wall then hit the ns bottom then detector without
history of ns entrance=18
{
    if (phforangrep[phangind].INCNSIND==nsind)
    {
        if (phforangrep[phangind].INCNSTYPE==3)
        {
            nsentnswall++; //phonon entered ns then hit the ns wall
            phivectotin_wi=phivectotin_wi+abs(phforangrep[phangind].IVEC);
            phjvectotin_wi=phjvectotin_wi+phforangrep[phangind].JVEC;
            phkvectotin_wi=phkvectotin_wi+phforangrep[phangind].KVEC;
        }
    }
}

```

```

}

if (phforangrep[phangind].INCNSTYPE==4)
{
    nsentnsext++; //phonon entered ns then hit the ns exit
}

if (phforangrep[phangind].INCNSTYPE==5)
{
    nsentnsbotdet++; //phonon entered ns then hit ns bottom then hit the det
}

if (phforangrep[phangind].INCNSTYPE==6)
{
    nsentnsbotsys++; //phonon entered then hit ns bottom then the sys boundary
}

if (phforangrep[phangind].INCNSTYPE==7)
{
    nsentnsexdet++; //phonon entered ns than hit the ns exit then det
}

if (phforangrep[phangind].INCNSTYPE==8)
{
    nsbotnsextdet++; //phonon originated from ns bottom then exited ns then hit the det
}

//////////

if (phforangrep[phangind].INCNSTYPE==9)
{
    nswallnsext_noenthist++; //no history of entering ns but interacted with ns wall then
exited=9
    phivectotout_wi=phivectotout_wi+abs(phforangrep[phangind].IVEC);
    phjvectotout_wi=phjvectotout_wi+phforangrep[phangind].JVEC;
    phkvectotout_wi=phkvectotout_wi+phforangrep[phangind].KVEC;
}

if (phforangrep[phangind].INCNSTYPE==10)
{
    nswallnsext_enthist++; //, history of entering ns and interaction with ns wall then
exited=10
    phivectotout_wi=phivectotout_wi+abs(phforangrep[phangind].IVEC);
    phjvectotout_wi=phjvectotout_wi+phforangrep[phangind].JVEC;
    phkvectotout_wi=phkvectotout_wi+phforangrep[phangind].KVEC;
}

if (phforangrep[phangind].INCNSTYPE==11)
{
    nsbotnsext++; //phonon that entered from open bottom and directly hit ns exit=11
}

if (phforangrep[phangind].INCNSTYPE==12)
{
    nswallnsbotsys_enthist++; // phonon with ns entrance history originated from ns wall
then exited from ns open bottom then hit sys boundary=12
}

if (phforangrep[phangind].INCNSTYPE==13)
{
    nswallnsbotsys_noenthist++; //phonon without ns entrance history originated from ns wall
then exited from ns open bottom then hit sys boundary=13
}

if (phforangrep[phangind].INCNSTYPE==14)
{

```

```

        nswallnsextnsdet_enthist++; //phonon originated from ns wall then exited ns then hit
        detector with ns entrance history=14
    }

    if (phforangrep[phangind].INCNSTYPE==15)
    {
        nswallnsextnsdet_noenthist++; //phonon originated from ns wall then exited ns then hit
        detector without ns entrance history=15
    }

    if (phforangrep[phangind].INCNSTYPE==16)
    {
        nsbotnswall++; //phonon originated from ns bottom then hit ns wall=16
        phivectotin_wi=phivectotin_wi+phforangrep[phangind].IVEC;
        phjvectotin_wi=phjvectotin_wi+phforangrep[phangind].JVEC;
        phkvectotin_wi=phkvectotin_wi+phforangrep[phangind].KVEC;
    }

    if (phforangrep[phangind].INCNSTYPE==17)
    {
        nswallnsbotnsdet_enthist++; //phonon originated from ns wall then hit the ns bottom then
        detector with history of ns entrance=17
    }

    if (phforangrep[phangind].INCNSTYPE==18)
    {
        nswallnsbotnsdet_noenthist++; // phonon originated from ns wall then hit the ns bottom
        then detector without history of ns entrance=18
    }

    if (phforangrep[phangind].INCNSTYPE==19)
    {
        nsentnsextnsdet++; // phonon originated from ns exit then hit the mesatop then detector
        with history of ns entrance=19
    }

    if (phforangrep[phangind].INCNSTYPE==20)
    {
        nswallnsextnsdet++; // phonon originated from ns exit then hit the mesatop then
        detector with history of ns entrance=19
    }

    if (phforangrep[phangind].INCNSTYPE==21)
    {
        nsbotnsextnsdet++; // phonon originated from ns exit then hit the mesatop then detector
        with history of ns entrance=19
    }

}

}

phangrepouts << nsind << "\t" <<countin<<"\t" <<countout<< "\t"
<<double(countout)/double(countin)<<"\t" << nsentnswall <<"\t" <<nsentnsextnsdet <<"\t"
<<nsentnsbotnsdet <<"\t" <<nsentnsbotnsys <<"\t" <<nsentnsextdet <<"\t" <<nsbotnsextdet<<"\t" <<
nswallnsextnsdet_noenthist<<"\t" <<nswallnsextnsdet_enthist<<"\t" <<nsbotnsextnsdet<<"\t"
<<nswallnsbotnsys_enthist<<"\t" <<nswallnsbotnsys_noenthist<<"\t"
<<nswallnsextnsdet_enthist<<"\t" <<nswallnsextnsdet_noenthist<<"\t" <<nsbotnswall<<"\t"
<<nswallnsbotnsdet_enthist<<"\t" <<nswallnsbotnsdet_noenthist<<"\t"<<
nsentnsextnsdet<<"\t"<< nswallnsextnsdet<<"\t"<< nsbotnsextnsdet <<"\t" <<
phivectotin/double(countin)<<"\t" <<phjvectotin/double(countin)<<"\t"
<<phkvectotin/double(countin)<< "\t" <<phivectotout/double(countout)<<"\t"
<<phjvectotout/double(countout)<<"\t" <<phkvectotout/double(countout)<<"\t"
<<phivectotin_wi/double(nsentnswall+nsbotnswall)<<"\t"
<<phjvectotin_wi/double(nsentnswall+nsbotnswall)<<"\t"
<<phkvectotin_wi/double(nsentnswall+nsbotnswall)<< "\t"
<<phivectotout_wi/double(nswallnsextnsdet_enthist+nswallnsextnsdet_noenthist)<<"\t"
<<phjvectotout_wi/double(nswallnsextnsdet_enthist+nswallnsextnsdet_noenthist)<<"\t"

```

```

<<pkvectotout_wi/double(nswallnsexthist+nswallnsexthist_noenthist)<< "\t"
<<double(nswallnsexthist)/double(nsentnswall)<< "\t"
<<double(nswallnsexthist+nswallnsexthist_noenthist)/double(nsentnswall+nsbotnswall)<< "\t"
<<double(nswallnsexthist+nsentnsexthist)/double(nsentnswall+nsentnsexthist)<< "\t"
<<double(nswallnsexthist+nswallnsexthist_noenthist+nsentnsexthist+nsbotnsexthist)/double(nsentnswall+n
sbotnswall+nsentnsexthist+nsbotnsexthist)<< endl;

countin_vec[nsind] =countin_vec[nsind]+countin;
countout_vec[nsind] =countout_vec[nsind]+countout;

nsentnsbotstsys_vec[nsind] =nsentnsbotstsys_vec[nsind]+nsentnsbotstsys;
nsentnswall_vec[nsind] =nsentnswall_vec[nsind]+nsentnswall;
nsentnsbotdet_vec[nsind] =nsentnsbotdet_vec[nsind]+nsentnsbotdet;
nsentnsexdet_vec[nsind] =nsentnsexdet_vec[nsind]+nsentnsexdet;
nsentnsexthist_vec[nsind] =nsentnsexthist_vec[nsind]+nsentnsexthist;
nsbotnsextdet_vec[nsind] =nsbotnsextdet_vec[nsind]+nsbotnsextdet;

nswallnsexthist_noenthist_vec[nsind] =nswallnsexthist_noenthist_vec[nsind]+nswallnsexthist_noenthist;
nswallnsexthist_vec[nsind] =nswallnsexthist_vec[nsind]+nswallnsexthist;
nsbotnsexthist_vec[nsind] =nsbotnsexthist_vec[nsind]+nsbotnsexthist;
nswallnsbotstsys_enthist_vec[nsind]
=nswallnsbotstsys_enthist_vec[nsind]+nswallnsbotstsys_enthist;
nswallnsbotstsys_noenthist_vec[nsind]
=nswallnsbotstsys_noenthist_vec[nsind]+nswallnsbotstsys_noenthist;
nswallnsextnsdet_enthist_vec[nsind]
=nswallnsextnsdet_enthist_vec[nsind]+nswallnsextnsdet_enthist;
nswallnsextnsdet_noenthist_vec[nsind]
=nswallnsextnsdet_noenthist_vec[nsind]+nswallnsextnsdet_noenthist;
nsbotnswall_vec[nsind] =nsbotnswall_vec[nsind]+nsbotnswall;
nswallnsbotnsdet_enthist_vec[nsind]
=nswallnsbotnsdet_enthist_vec[nsind]+nswallnsbotnsdet_enthist;
nswallnsbotnsdet_noenthist_vec[nsind]
=nswallnsbotnsdet_noenthist_vec[nsind]+nswallnsbotnsdet_noenthist;
nsentnsextnsdet_vec[nsind]=nsentnsextnsdet_vec[nsind]+nsentnsextnsdet;
nswallnsextnsdet_vec[nsind]=nswallnsextnsdet_vec[nsind]+nswallnsextnsdet;
nsbotnsextnsdet_vec[nsind]=nsbotnsextnsdet_vec[nsind]+nsbotnsextnsdet;
phivectotout_vec[nsind]=phivectotout_vec[nsind]+phivectotout;
phjvectotout_vec[nsind]=phjvectotout_vec[nsind]+phjvectotout;
phkvectotout_vec[nsind]=phkvectotout_vec[nsind]+phkvectotout;
phivectotin_vec[nsind]=phivectotin_vec[nsind]+phivectotin;
phjvectotin_vec[nsind]=phjvectotin_vec[nsind]+phjvectotin;
phkvectotin_vec[nsind]=phkvectotin_vec[nsind]+phkvectotin;
phivectotout_wi_vec[nsind]=phivectotout_wi_vec[nsind]+phivectotout_wi;
phjvectotout_wi_vec[nsind]=phjvectotout_wi_vec[nsind]+phjvectotout_wi;
phkvectotout_wi_vec[nsind]=phkvectotout_wi_vec[nsind]+phkvectotout_wi;
phivectotin_wi_vec[nsind]=phivectotin_wi_vec[nsind]+phivectotin_wi;
phjvectotin_wi_vec[nsind]=phjvectotin_wi_vec[nsind]+phjvectotin_wi;
phkvectotin_wi_vec[nsind]=phkvectotin_wi_vec[nsind]+phkvectotin_wi;

nsentnsbotstsys=0;
nsentnswall=0;
nsentnsbotdet=0;
nsentnsexdet=0;
nsentnsexthist=0;
nsbotnsextdet=0;

nswallnsexthist_noenthist=0; //
nswallnsexthist=0; //
nsbotnsexthist=0; //
nswallnsbotstsys_enthist=0; //
nswallnsbotstsys_noenthist=0; //
nswallnsextnsdet_enthist=0; //
nswallnsextnsdet_noenthist=0; //
nsbotnswall=0; //
nswallnsbotnsdet_enthist=0; //
nswallnsbotnsdet_noenthist=0; //

```

```

nsentnsextmtnsdet=0;
nswallnsextmtnsdet=0;
nsbotnsextmtnsdet=0;

}
if (ifangrepbin==1)
{
    binlrec=1;
    for(binind=0;binind<(mxsize*mzsize);binind++)
    {
        binl=binind/mxsize;
        if (binl > binlrec)
        {
            phangrepbinin << endl;
            phangrepbinout << endl;
        }
        phangrepbinin << bincountin[binind]<<"\t";
        phangrepbinout << bincountout[binind]<<"\t";

        bincountin[binind]=0;
        bincountout[binind]=0;

        binlrec=binl;
    }
}
if (ifangrepdet==1)
{
    phangrepout.close();
    if (ifangrepbin==1)
    {
        phangrepbinin.close();
        phangrepbinout.close();
    }
}

phangrepouts << endl <<endl << "Phonons originating from generator and then hitting detector
after passing beneath the nanosheet:" <<endl << detcountundns << endl << endl << "Phonons
originating from ns gaps and then hitting detector:" <<endl << detcountfrgp <<endl;

phangrepouts.close();

phforangrep.clear();
}

if (ifdetrep==1)
{
    if (ifdetrepbin==1)
    {
        detbinlrec=1;
        for(detbinind=0;detbinind<(mdetxsize*mdetzsize);detbinind++)
        {
            detbinl=detbinind/mdetxsize;
            if (detbinl > detbinlrec)
            {
                phdetrepbinout << endl;
            }
            phdetrepbinout << detbincount[detbinind]<<"\t";

            detbincount[detbinind]=0;

            detbinlrec=detbinl;
        }
        phdetrepbinout.close();
    }
    phdetrepout.close();
}

```



```

    transmission << frequency << "\t" << ((4.135667516*pow(double(10),double(-
9))*frequency)+twodelta) << "\t" << syscount << "\t" << detcount << "\t" <<
double(syscount)/double(syscount+detcount) << "\t" <<
double(detcount)/double(syscount+detcount) << endl;
    if ((syscount+detcount) != (phononsperf+1))
    {
        cout << "PROBLEM!!!. There are some phonons which are not detected nor captured by the
system borders" << endl;
        cout<< "syscount+detcount= "<<(syscount+detcount)<< " Phonons per frequency:" <<
phononsperf<< endl;
        /*system ("pause");*/
    }
    freqcount=freqcount+1;
    syscounttot=syscount+syscounttot;
    detcounttot=detcount+detcounttot;
}

string freqangfnsoverall =angfilename+"_overallshort.txt";
phangrepoutsov.open(freqangfnsoverall.c_str(), ios::app);

phangrepoutsov << "Overall short report for phonon angle and count for nanosheets (summed and
averaged for all frequencies)" << endl;
phangrepoutsov << "NS index" << "\t" << "#phonons entering the ns"<<"\t" << "#phonons exiting
the ns"<<"\t" << "#the ratio of phonons exiting/entering the ns"<<"\t" << "#phonons entering
the ns then hitting the ns wall" <<"\t" << "#phonons entering the ns then hitting the ns exit"
<<"\t" << "#phonons entering the ns then hitting img ns bottom then hitting the det" <<"\t"
<< "#phonons entering the ns then hitting img bottom then hitting the sys boundary" <<"\t"
<< "#phonons entering the ns then hitting the ns exit then detector" <<"\t" << "#phonons
originating from img ns bottom then exiting from ns then hitting the detector" <<"\t"
<< "#phonons originating from ns wall then exiting from ns without history of entering ns"
<<"\t" << "#phonons originating from ns wall then exiting from ns with history of entering ns"
<<"\t" << "#phonons originating from ns bottom then exiting from ns" <<"\t" << "#phonons
originating from ns wall then exiting from img ns bottom then hitting sys boundary with ns
enterance history" <<"\t" << "#phonons originating from ns wall then exiting from img ns bottom
then hitting sys boundary without phonon enterance history " <<"\t" << "#phonons originating
from ns wall then exiting from ns then hitting detector with phonon enterance history " <<"\t"
<< "#phonons originating from ns wall then exiting from ns then hitting detector without phonon
enterance history " <<"\t" << "#phonons originating from img ns bottom then hitting ns wall"
<<"\t" << "#phonons originating from ns wall then hitting img ns bottom then hitting detector
with phonon enterance history " <<"\t" << "#phonons originating from ns wall then hitting img
ns bottom then hitting detector without phonon enterance history " <<"\t" << "Phonons
originating from ns ent hittinh ns exit and then hitting mesatop then detector" <<"\t" <<
"Phonons originating from ns wall then hitting ns exit and then hitting mesatop then detector"
<<"\t" << "Phonons originating from ns bot then hitting ns exit and then hitting mesatop then
detector" <<"\t" << "Entering phonon i vector average"<<"\t" << "Entering phonon j vector
average"<<"\t" << "Entering phonon k vector average"<< "\t" << "Exiting phonon i vector
average"<<"\t" << "Exiting phonon j vector average"<<"\t" << "Exiting phonon k vector
average"<<"\t" << "Wall Interacting Entering phonon i vector average"<<"\t" << "Wall Interacting
Entering phonon j vector average"<<"\t" << "Wall Interacting Entering phonon k vector
average"<< "\t"<< "Wall Interacting Exiting phonon i vector average"<<"\t" << "Wall Interacting
Exiting phonon j vector average"<<"\t" << "Wall Interacting Exiting phonon k vector average"<<
"\t" << "Wall interacting ns phonon transmission for closed system"<<"\t" << "Wall interacting
ns phonon transmission for open system"<<"\t" << "Total ns phonon transmission for closed
system"<<"\t" << "Total ns phonon transmission for open system"<< endl;

for (nsind=1; nsind <= howmanyys; nsind++)
{
    phangrepoutsov << nsind << "\t" << countin_vec[nsind]<<"\t" << countout_vec[nsind]<< "\t"
<< double(countout_vec[nsind])/double(countin_vec[nsind])<<"\t" << nsentnswall_vec[nsind]
<< "\t" << nsentnsext_vec[nsind] << "\t" << nsentnsbotdet_vec[nsind] << "\t"
<< nsentnsbotsys_vec[nsind] << "\t" << nsentnsexdet_vec[nsind] << "\t"

```

```

<<nsbotnsextdet_vec[nsind]<<"\t" << nswallnsext_noenthist_vec[nsind]<<"\t"
<<nswallnsext_enthist_vec[nsind]<<"\t" <<nsbotnsext_vec[nsind]<<"\t"
<<nswallnsbotsys_enthist_vec[nsind]<<"\t" <<nswallnsbotsys_noenthist_vec[nsind]<<"\t"
<<nswallnsextnsdet_enthist_vec[nsind]<<"\t" <<nswallnsextnsdet_noenthist_vec[nsind]<<"\t"
<<nsbotnswall_vec[nsind]<<"\t" <<nswallnsbotnsdet_enthist_vec[nsind]<<"\t"
<<nswallnsbotnsdet_noenthist_vec[nsind]<<"\t" << nsentnsextmtnsdet_vec[nsind]<<"\t" <<
nswallnsextmtnsdet_vec[nsind]<<"\t"<< nsbotnsextmtnsdet_vec[nsind]<<"\t" <<
phivectotin_vec[nsind]/double(countin_vec[nsind])<<"\t"
<<phjvectotin_vec[nsind]/double(countin_vec[nsind])<<"\t"
<<phkvectotin_vec[nsind]/double(countin_vec[nsind])<< "\t"
<<phivectotout_vec[nsind]/double(countout_vec[nsind])<<"\t"
<<phjvectotout_vec[nsind]/double(countout_vec[nsind])<<"\t"
<<phkvectotout_vec[nsind]/double(countout_vec[nsind])<<"\t"
<<phivectotin_wi_vec[nsind]/double(nsentnswall_vec[nsind]+nsbotnswall_vec[nsind])<<"\t"
<<phjvectotin_wi_vec[nsind]/double(nsentnswall_vec[nsind]+nsbotnswall_vec[nsind])<<"\t"
<<phkvectotin_wi_vec[nsind]/double(nsentnswall_vec[nsind]+nsbotnswall_vec[nsind])<< "\t"
<<phivectotout_wi_vec[nsind]/double(nswallnsext_enthist_vec[nsind]+nswallnsext_noenthist_vec[n
sind])<<"\t"
<<phjvectotout_wi_vec[nsind]/double(nswallnsext_enthist_vec[nsind]+nswallnsext_noenthist_vec[n
sind])<<"\t"
<<phkvectotout_wi_vec[nsind]/double(nswallnsext_enthist_vec[nsind]+nswallnsext_noenthist_vec[n
sind])<< "\t" <<double(nswallnsext_enthist_vec[nsind])/double(nsentnswall_vec[nsind])<<"\t"
<<double(nswallnsext_enthist_vec[nsind]+nswallnsext_noenthist_vec[nsind])/double(nsentnswall_v
ec[nsind]+nsbotnswall_vec[nsind])<<"\t"
<<double(nswallnsext_enthist_vec[nsind]+nsentnsext_vec[nsind])/double(nsentnswall_vec[nsind]+n
sentnsext_vec[nsind])<<"\t"
<<double(nswallnsext_enthist_vec[nsind]+nswallnsext_noenthist_vec[nsind]+nsentnsext_vec[nsind]
+nsbotnsext_vec[nsind])/double(nsentnswall_vec[nsind]+nsbotnswall_vec[nsind]+nsentnsext_vec[ns
ind]+nsbotnsext_vec[nsind])<< endl;
}

phangrepoutsov << endl <<endl << "Phonons originating from generator and then hitting detector
after passing beneath the nanosheet:" <<endl << detcountundns << endl << endl << "Phonons
originating from ns gaps and then hitting detector:" <<endl << detcountfrgp <<endl;

transmission << "-----SUMMED-WEIGHTED AVERAGED FOR
ALL FREQUENCIES DEPENDING ON EMISSION SPECTRA-----"
----" << endl;
transmission << "Escape to System Border count" << "\t" << "Detected count" << "\t" << "Escape
to system border ratio" << "\t" << "Detected count ratio"<< endl;
transmission << syscountot << "\t" << detcounttot << "\t" <<
double(syscountot)/double(syscountot+detcountot) << "\t" <<
double(detcountot)/double(syscountot+detcountot) << endl;

cout << "Total number of utilized phonons: " << syscountot+detcountot <<endl;
system ("pause");
return 0;
}

```

## D.7. Supplementary Methods: Analytical Model Matlab Code

### GEOMFRACTIONDIF.m

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Calculation of approximate fractional solid angle subtended by detector
% relative to segment of generator, considered as a point source at
% arbitrary location on sidewall. Detector is on opposite sidewall only.
% This segment of generator is assumed to emit at a uniform rate.
% This function should be suitable for use in a numerical integral over
% entire generator and entire detector. Can also be expanded to
% incorporate phonon focus factor and acoustic mismatch factor as a
% function of angle, but in simplest approximation the emitter segment
% is assumed to emit entirely isotropically. Sidewall angle is 54.7
% degrees for KOH-etched Si.
%
% Coordinate system is:
% x: parallel to the mesa sidewall (+x pointing to the right)
% y: pointing across the mesa (+y pointing towards the detector sidewall)
% z: pointing out of the plane of the chip
% (0,0,0) is taken to be the midpoint of the emitter sidewall at midpoint
% of the emitter. So for mesa of height h, the top of mesa is at z=h/2.
% Note that the segment of emitter being considered is not necessarily at
% (0,0,0)!
%
% J Hertzberg 5-3-2012
% Based in part on 'GeomFraction.m'
% Modified 6-16-12 by J Hertzberg to include cosine (Lambert law)
% emission distribution and to include phonon focus factors in (100)
% plane, from published values
%
% Arguments
%
% Ar      (x,y,z) position of emitter segment (3-value column vector)
% Bs      Area of detector on sidewall (actual area, not projected
%        area as seen from above)
% Br      (x,y,z) position of detector (3-value column vector)
% Ll      Mesa width, from midpoint of one sidewall to other
% h       Height of mesa
% ThetaEmt Emitter sidewall angle relative to horizontal (degrees)
% ThetaDet Detector sidewall angle relative to horizontal (degrees)
%
% Output
% Fract    Fraction of phonon emission from generator segment that is
%        expected to reach the detector
%
%
% Mahmut's additions:
%
% 1) ifref: boolean value to include reflective top
%    surface (1=yes, 0=no)
%
% 2) Corrected cosine distribution algorithm
%
% 3) Incorporated PFF from Monte Carlo Simulations
%
% 4) Fixed phonon angle calculation mistake
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Fract = GeomFractionDif(Ar, Bs, Br, Ll, h, ThetaEmt,
ThetaDet,Segwidxdet,Segwidydet, ifref)

%Find the vector pointing from emitter to detector (line-of-sight vector).
%Find its length and unit vector.
rvec = Br - Ar;
```

```

rmag = norm(rvec);
runit = rvec/rmag;

%Find the angle of the line-of-sight vector relative to the normal of
%the emitter sidewall. This may be useful for e.g. angle-dependent acoustic
%transmission factors
Aunit = [0; sin(ThetaEmit*pi/180); -cos(ThetaEmit*pi/180)];
Emitangl = 180 / pi * acos(abs(Aunit' * runit));

%Find the angle of the line-of-sight vector relative to the normal of
%the detector sidewall. This may be useful for e.g. angle-dependent acoustic
%transmission factors
Bunit = [0; -sin(ThetaDet*pi/180); -cos(ThetaDet*pi/180)];
Detangl = 180 / pi * acos(abs(Bunit' * runit));

%Find the angle of the line-of-sight vector relative to the y direction.
%This may be useful for e.g. phonon focus relative to this direction of
%crystal. Phix is in xy plane (i.e. lies within the (100) plane in our
%typical chip, with phix = 0 being in (110) direction, phix=45deg being in
%(100) direction). Phiz is angle in yz plane
rmx=(rvec.*[1;1;0]);
rmxmag = norm(rmx);
rmxunit=rmx/rmxmag;

Detanglx = 180 / pi * acos(abs([0;-1;0]' * rmxunit));
xdegwidth=180*atan((cos(Detanglx*pi/180)*Segwidxdet/2)/rmxmag)/pi;

rmz=(rvec.*[0;1;1]);
rmzmag = norm(rmz);
rmzunit=rmz/rmzmag;

Detanglz = 180 / pi * acos(abs(Bunit' * rmzunit));
zdegwidth=180*atan((cos(Detanglz*pi/180)*Segwidzdet/2)/rmzmag)/pi;

Phix = 180 / pi * acos([0;1;0]' * rmxunit);
Phiz = 180 / pi * acos([0;1;0]' * rmzunit);

%Acoustic transmission factor from emitter into material.
%Option 1: For isotropic emission, set the emitter transmission factor to 1
%Tfactemt = 1;
%Option 2: Cosine (Lambert law) distribution of emitted flux. Neglect
%acoustic mismatch transmission factor or assume it is not angle dependent.
Tfactemt = 2*cos(Emitangl*pi/180);
%Option 3: assumption of isotropic phonon flux within the emitter; emission
%will be the intercepted boundary area, multiplied by an acoustic mismatch
%factor.
%[Rfrac, Tfracemt, Thlemt] = BoundaryTransTHoutfromTHin(3.26, 2.73, 5.1, 2.33,
Emitangl);
%Tfactemt = cos(pi/180.*Thlemt) .* Tfracemt;

%Acoustic transmission factor from material into detector.
%Option 1: transmission factor is independent of angle and is unity (this
%is reasonable approximation up to about 60 degrees of incidence and for
%mismatch between Al and Si, which has >95% mismatch at normal incidence)
Tfactdet = 1;
%Option 2: calculated angle-dependent acoustic mismatch factor
%[Rfrac, Tfactdet, Thldet] = BoundaryTransTHintoTHout(5.1, 2.33, 3.26, 2.73, Detangl);

%Mahmut's MC calculated 3D pf factors are read from the workspace variable
%pff110

persistent pff;
if isempty(pff)
pff=evalin('base', 'pff110');
disp 'Writing pff variable ...'

```

```

end

persistent xdeg;
if isempty(xdeg)
    xdeg=180*atan(((0.5:1:999.5)*10)-5000)/3000)/pi;
    disp 'Writing xdeg variable ...'
end

% Determine which pff factor to pick based on emission angle
xlow=find(xdeg>(Phix-xdegwidth),1,'first');
xhigh=find(xdeg<(Phix+xdegwidth),1,'last');

if isempty(xlow)
    xlow=xhigh;
elseif isempty (xhigh)
    xhigh=xlow;
elseif isempty(xlow)
    disp 'both xlow and xhigh empty!!!'
end

if (xhigh<xlow)
    xlow=xhigh;
end

zlow=find(xdeg>(Phiz-zdegwidth),1,'first');
zhigh=find(xdeg<(Phiz+zdegwidth),1,'last');

if isempty(zlow)
    zlow=zhigh;
elseif isempty (xhigh)
    zhigh=zlow;
elseif isempty(zlow)
    disp 'both zlow and zhigh empty!!!'
end

if (zhigh<zlow)
    zlow=zhigh;
end

% End determine which pff factor to pick based on emission angle

%Pick the PFF factor
Phfocfact = mean(mean(pff(zlow:1:zhigh,xlow:1:xhigh))');
if isnan(Phfocfact)==1
    disp 'nan focus factor'
    Phix
    Phiz
    xlow
    xhigh
    zlow
    zhigh
    xdegwidth
    zdegwidth
end

%The area exposed to the flux is the detector area perpendicular to the
%line-of-sight vector.
Bsperp = abs(Bs * Bunit' * runit);
if Bunit' * runit > 0
    Bsperp = 0;
end

%Flux will be modified by phonon focus and acoustic transmission factors.
%Fraction collected is area exposed to flux / 4 pi r^2

```

```

%add factor of two for approximate reflective mesa top: i.e. :Fract = 2*2*Tfactemt *
Tfactdet * Phfocfact * Bsperp / rmag^2 / 4 / pi; OR add GeomFractionDifref(Ar, Bs, Br,
L1, h, ThetaEmt, ThetaDet,Segwidxdet,Segwidydct) for exact solution;

if (ifref==1)
    refadd=GeomFractionDifref(Ar, Bs, Br, L1, h, ThetaEmt,
ThetaDet,Segwidxdet,Segwidydct);
else
    refadd=0;
end

Fract = (2*Tfactemt * Tfactdet * Phfocfact * Bsperp / rmag^2 / 4 / pi)+refadd;

end

```

### GEOMFRACTIONINTGTEMIT.m (Calls GEOMFRACTIONDIF.m)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%   Integrated calculation of approximate fractional solid angle subtended
%   by detector relative to generator. Integration is over generator.
%   Detector is considered a point. Generator is considered to be a
%   rectangle lying on the sidewall. The x-coordinate of the middle of this
%   rectangle is zero but the generator y midpoint may be higher or lower
%   than zero. Detector may be either full detector, or a segment of
%   detector, for use in a further integration, as needed. Detector
%   position is at arbitrary location on sidewall. Detector is on opposite
%   sidewall only. All of generator is assumed to emit at a uniform rate,
%   and is divided into equal-size segments that emit equal fluxes. The
%   fluxes may be modified by incorporating into this function the phonon
%   focus factor and acoustic mismatch factor as a function of angle.
%   In simplest approximation each emitter segment is assumed to emit
%   entirely isotropically. In KOH-etched Si, the sidewall angle is 54.7
%   degrees.
%
%   Coordinate system is:
%   x: parallel to the mesa sidewall (+x pointing to the right)
%   y: pointing across the mesa (+y pointing towards the detector sidewall)
%   z: pointing out of the plane of the chip
%   (0,0,0) is taken to be the midpoint of the emitter sidewall at the
%   x-coordinate midpoint of the emitter. So for mesa of height h, the top
%   of mesa is at z=h/2. Note that the emitter midpoint is not necessarily
%   at (0,0,0)!
%
%   J Hertzberg 5-5-2012
%   Based in part on 'GeomFractionDif.m'
%
%   Arguments
%
%   Emitwid      Width of emitter (microns)
%   Emitylow     y coordinate of lower edge of emitter rectangle
%   Emityhi      y coordinate of upper edge of emitter rectangle
%   Bs           Area of detector or detector segment on sidewall (actual
%                area, not projected area as seen from above)
%   Br           (x,y,z) position of detector or detector segment (3-value
%                column vector)
%   L1           Mesa width, from midpoint of one sidewall to other
%   h            Height of mesa
%   ThetaEmt     Emitter sidewall angle relative to horizontal (degrees)
%   ThetaDet     Detector sidewall angle relative to horizontal (degrees)
%   Stepsx       Number of segments to divide the emitter into, in x
%                dimension
%   Stepsy       Number of segments to divide the emitter into, in y
%                dimension
%
%   Output
%   Fract        Fraction of phonon emission from generator that is
%                expected to reach the detector or detector segment at
%                position Br
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

% Mahmut's addition: ifref: passes the boolean value to GeomFractionDif to
% include reflective top surface
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

function Fract = GeomFractionIntgtEmit(Emitwid, Emitylow, Emityhi, Bs, Br, L1, h,
ThetaEmit, ThetaDet, Stepsx, Stepsy, Segwidxdet, Segwidydet, ifref)

clear Fract;
Fract = 0;

%Divide the emitter up into segments of equal area. The flux from each
%segment is equal to that segment's fraction of the whole emitter area.
Segfract = 1/(Stepsx*Stepsy);
Segwidx = Emitwid/Stepsx;
Segwidy = (Emityhi - Emitylow)/Stepsy;

for mx = 0:(Stepsx-1)
    % Integrate in x direction
    thisx = -(Emitwid - Segwidx)/2 + mx*Segwidx;
    for my = 0:(Stepsy-1)
        %Integrate in y direction. Z follows from y.
        thisy = Emitylow + Segwidy/2 + my*Segwidy;
        thisz = thisy * tan(ThetaEmit*pi/180);
        thisAr = [thisx; thisy; thisz]; %Construct the vector defining the position of
this emitter segment
        thisfract = Segfract * GeomFractionDif(thisAr, Bs, Br, L1, h, ThetaEmit,
ThetaDet, Segwidxdet, Segwidydet, ifref);
        Fract = Fract + thisfract;
    end
end

end

```

## GEOMFRACTIONINTGTEMITINTGTDET.m (Calls GEOMFRACTIONINTGTEMIT.m)

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Integrated calculation of approximate fractional solid angle subtended
% by detector relative to emitter. Integration is over detector.
% Additional integration over emitter is included via the function
% GeomFractionIntgtEmit. Emitter and detectors are both considered to be
% rectangles lying on the sidewall. The x-coordinate of the middle of the
% emitter rectangle is zero but its y midpoint may be higher or lower
% than zero. Detector position is at arbitrary location on sidewall.
% Detector is on opposite sidewall only. All of generator is assumed to
% emit at a uniform rate, and is divided into equal-size segments that
% emit equal fluxes. The fluxes may be modified by incorporating into
% this function the phonon focus factor and acoustic mismatch factors as
% a function of angle. In simplest approximation each emitter segment is
% assumed to emit entirely isotropically. In KOH-etched Si, the sidewall
% angle is 54.7 degrees.
%
% Coordinate system is:
% x: parallel to the mesa sidewall (+x pointing to the right)
% y: pointing across the mesa (+y pointing towards the detector sidewall)
% z: pointing out of the plane of the chip
% (0,0,0) is taken to be the midpoint of the emitter sidewall at the
% x-coordinate midpoint of the emitter. So for mesa of height h, the top
% of mesa is at z=h/2. Note that the emitter midpoint is not necessarily
% at (0,0,0)!
%
% J Hertzberg 5-9-2012
% Based in part on 'GeomFractionDif.m' and 'GeomFractionIntgtEmit.m'
%
% Arguments
%
% Emitwid      Width of emitter (microns)

```

```

% Emitylow    y coordinate of lower edge of emitter rectangle. Will be
%             negative value. (microns)
% Emityhi    y coordinate of upper edge of emitter rectangle. Will be
%             positive value. (microns)
% Detxmids    x coordinate of midpoint of detector, i.e. lateral offset
%             of detector from emitter-detector midline (microns)
% Detwid      Width of detector (microns)
% Detylow     y coordinate of lower edge of detector rectangle, relative
%             to midpoint of detector mesa. Will be positive. (microns)
% Detyhi      y coordinate of upper edge of detector rectangle, relative
%             to midpoint of detector mesa. Will be negative. (microns)
% L1          Mesa width, from midpoint of one sidewall to other
% h           Height of mesa
% ThetaEmit   Emitter sidewall angle relative to horizontal (degrees)
% ThetaDet    Detector sidewall angle relative to horizontal (degrees)
% StepsEmtxy  Number of segments to divide the emitter into
%             [x steps; y steps]
% StepsDetxy  Number of segments to divide the detector into
%             [x steps; y steps]
%
% Output
% Fract       Fraction of phonon emission from generator that is
%             expected to reach the detector
%
% Mahmut's addition: ifref: passes the boolean value to GeomFractionDif to include
% reflective top surface
%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
function Fract = GeomFractionIntgtEmitIntgtDet(Emitwid, Emitylow, Emityhi, Detxmids,
Detwid, Detylow, Detyhi, L1, h, ThetaEmit, ThetaDet, StepsEmtxy, StepsDetxy, ifref)

clear Fract;
Fract = 0;

%Divide the detector up into segments of equal area. Determine the x and y
%step sizes and the real area of each segment lying on the sidewall.
Segwidx = Detwid/StepsDetxy(1);
Segwidy = (Detylow - Detyhi)/StepsDetxy(2);
SegBs = Segwidx*Segwidy/cos(ThetaDet*pi/180);

for mx = 0:(StepsDetxy(1)-1)
    %Integrate in x direction

    thisx = Detxmids - (Detwid - Segwidx)/2 + mx*Segwidx;
    for my = 0:(StepsDetxy(2)-1)
        %Integrate in y direction. Z follows from y.
        thisy = L1 + Detyhi + Segwidy/2 + my*Segwidy;
        thisz = -(Detyhi + Segwidy/2 + my*Segwidy) * tan(ThetaDet*pi/180);
        thisBr = [thisx; thisy; thisz]; %Construct the vector defining the position of
this detector segment
        thisfract = GeomFractionIntgtEmit(Emitwid, Emitylow, Emityhi, SegBs, thisBr,
L1, h, ThetaEmit, ThetaDet, StepsEmtxy(1), StepsEmtxy(2), Segwidx, Segwidy, ifref);
        Fract = Fract + thisfract;
    end
end
end

```

## GEOMFRACTNSMOD6.m

(Calls GEOMFRACTIONINTGTemitINTGTDET.m and calculates the ballistic phonon transport through the Si nanosheets)

%This algorithm calls phonon transmission calculations functions with



```

%appropriate geometric definition of the nanosheet phonon spectrometry device to
calculate the phonon transmission
%through Si nanosheets.
%Written by MAHMUT AKSIT.
%
% The inputs are:
% Chipname: in order to find nanosheet geometry details in workspace.
% asd: step size in emitter along x
% asd2: step size in emitter long z
% distons: emitter to nanosheet distance

function geomfractnsmod6=geomfractnsmod6(chipname,asd,asd2,distons)

% extract nanosheet and mesa details from workspace and produce necessary variable to
form the transpot medium:
Afractname=strcat(chipname, '_Afract');
Aswname=strcat(chipname, '_Asw');
Bswname=strcat(chipname, '_Bsw');
Bshname=strcat(chipname, '_Bsh');
sidewall_widthname=strcat(chipname, '_Sw');
nspname=strcat(chipname, '_nsp');
nslname=strcat(chipname, '_nsl');
nswname=strcat(chipname, '_nsw');
nsdname=strcat(chipname, '_nsd');

Afract = evalin('base', Afractname);
Asw = evalin('base', Aswname);
Bsw = evalin('base', Bswname);
Bsh = evalin('base', Bshname);
sw = evalin('base', sidewall_widthname);
nspitch=evalin('base', nspname);
nslength=evalin('base', nslname);
nswidth=evalin('base', nswname);
nsdepth=evalin('base', nsdname);

calclim=8.3;
mesalength=7;
freq=400000000000;
rough=2;
h=0.8;
nsThdet=89.9;
ThetaEmt=54.74;
ThetaDet=54.74;

nsdetylow=(nsdepth-(h/2))/tan(nsThdet*pi/180);
nsdetyhigh=-h/2/tan(nsThdet*pi/180);

nsemtylow=-(nsdepth-(h/2))/tan(nsThdet*pi/180);
nsemyhigh=h/2/tan(nsThdet*pi/180);

Emitylow=-sw/2;
Emityhigh=sw/2;

Detylow=sw/2;
Detyhigh=sw/2-Bsh;

emtonsL=distons+(sw/2);
nstodetL= mesalength - distons - nslength+(sw/2); %nsemyhigh+
k=0;
transmat=zeros(1,15);
ballistic_int=0;
ballistic_ns_int=0;
ballistic_of_int=0;

while ((k*nspitch)+nspitch/2-nswidth/2) < calclim)
    % emitter to nanosheet entrance tramission

```

```

    emtons_int=GeomFractionIntgtEmitIntgtDet(Asw, Emitylow, Emityhigh,
    nspitch/2+(k*(nspitch)), nswidth, nsdetylow, nsdetyhigh, emtonsL, h, ThetaEmt,
    nsThdet, [20;10], [10;10],1);

    %Transmission through the Nanosheets
    for p=(-(asd/2)+0.5):1:(asd/2)-0.5

        %first define the geometry
        baldetmax = min(Bsw/2, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
        (nspitch/2+(k*(nspitch))))*(nstodetL+((sw-Bsh)/2))/(emtonsL+nslength)) +
        (nswidth*(mesalength+sw+((sw-Bsh)/2))/(emtonsL+nslength)/2));
        baldetmin = max(-Bsw/2, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
        (nspitch/2+(k*(nspitch))))*(nstodetL+((sw-Bsh)/2))/(emtonsL+nslength)) -
        (nswidth*(mesalength+sw+((sw-Bsh)/2))/(emtonsL+nslength)/2));

        baldetmax2 = min(Bsw/2, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
        (nspitch/2+(k*(nspitch))))*(nstodetL+((sw-Bsh)/2)+nslength)/(emtonsL)) +
        (nswidth*(mesalength+sw+((sw-Bsh)/2))/(emtonsL)/2));
        baldetmin2 = max(-Bsw/2, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
        (nspitch/2+(k*(nspitch))))*(nstodetL+((sw-Bsh)/2)+nslength)/(emtonsL)) -
        (nswidth*(mesalength+sw+((sw-Bsh)/2))/(emtonsL)/2));
        p;

        if (baldetmax2 <= baldetmin)|| (baldetmin2>=baldetmax)

            ballistic_int_temp=0;

        else

            baldeteffmax=min(baldetmax,baldetmax2);
            baldeteffmin=max(baldetmin,baldetmin2);

            baldeteff = max(0, baldeteffmax - baldeteffmin);
            baldeteff_x = (baldeteffmax + baldeteffmin)/2 - (p*(Asw/asd));

            for pz=0.5:1:(asd2-0.5)

                pz;
                baldetminz =0+((sw-Bsh)*tan(ThetaDet*pi/180));
                baldetmaxz=h;

                baldetminz2 =max(((sw-Bsh)*tan(ThetaDet*pi/180)), (0-((pz*(h/asd2))-
                0)*((nstodetL+nslength+((sw-Bsh)/2))/(emtonsL-
                (sw/2)+((pz*(h/asd2))/tan(ThetaEmt*pi/180))))));
                baldetmaxz2= min(h, (nsdepth-((pz*(h/asd2))-
                nsdepth)*((nstodetL+nslength+((sw-Bsh)/2))/(emtonsL-
                (sw/2)+((pz*(h/asd2))/tan(ThetaEmt*pi/180))))));

                baldetmaxz2=max(baldetmaxz2,((sw-Bsh)*tan(ThetaDet*pi/180)));

                if (baldetmaxz2 <= baldetminz)|| (baldetminz2>=baldetmaxz) %2nd
statement may not be possible but I will keep it there just in case.

                    ballistic_int_temp=0;
                    %fprintf(1,'didnt integrate due to z restriction\n');
                    %pause

            else

                baldeteffzmax=min(baldetmaxz,baldetmaxz2);
                baldeteffzmin=max(baldetminz,baldetminz2);
                %integrate transmission through the nanosheets

            ballistic_int_temp=(1/(asd*(asd2)))*GeomFractionIntgtEmitIntgtDet(Asw/(asd), -
            ((pz*(h/asd2))/tan(ThetaEmt*pi/180))+((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2, -
            ((pz*(h/asd2))/tan(ThetaEmt*pi/180))-((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2,
            baldeteff_x, baldeteff, -((sw/2)-(baldeteffzmax/tan(ThetaDet*pi/180)))-((sw/2)-
            (baldeteffzmin/tan(ThetaDet*pi/180))), mesalength+sw, h, ThetaEmt, ThetaDet, [1;1],
            [20;10],1);

```

```

        %pause
    end

    ballistic_int=ballistic_int_temp+ballistic_int;

end

end

    balnsmax =(nspitch/2+(k*(nspitch)))+(nswidth/2);
    balnsmin =(nspitch/2+(k*(nspitch)))-(nswidth/2);

    baldetmax2 = min(balnsmax, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
(nspitch/2+(k*(nspitch))))*(nslength)/(emtonsL)) +
(nswidth*(emtonsL+nslength)/(emtonsL)/2));
    baldetmin2 = max(balnsmin, (nspitch/2+(k*(nspitch))-((p*(Asw/asd))-
(nspitch/2+(k*(nspitch))))*(nslength)/(emtonsL)) -
(nswidth*(emtonsL+nslength)/(emtonsL)/2));

    if (baldetmax2 <= balnsmin)|| (baldetmin2>=balnsmax)

        ballistic_ns_int_temp=0;
        %fprintf(1,'didn't integrate due to x restriction for ns\n');

    else

        baldeteffmax=min(balnsmax,baldetmax2);
        baldeteffmin=max(balnsmin,baldetmin2);

        baldeteff = max(0, baldeteffmax - baldeteffmin);
        baldeteff_x = (baldeteffmax + baldeteffmin)/2 - (p*(Asw/asd));

        for pz=0.5:1:(asd2-0.5)

            pz;
            balnsminz = 0;
            balnsmaxz= nsdepth;

            baldetminz2 = max(balnsminz, (0-((pz*(h/asd2))-
0)*(nslength)/(emtonsL-(sw/2)+(pz*(h/asd2))/tan(ThetaEmt*pi/180)))));
            baldetmaxz2 = min(balnsmaxz, (nsdepth-((pz*(h/asd2))-
nsdepth)*(nslength)/(emtonsL-(sw/2)+(pz*(h/asd2))/tan(ThetaEmt*pi/180)))));

            if (baldetmaxz2 <= balnsminz)|| (baldetminz2>=balnsmaxz) %2nd statement
may not be possible but I will keep it there just in case.

                ballistic_ns_int_temp=0;
                %fprintf(1,'didn't integrate due to z restriction for ns\n');

            else

                baldeteffzmax=min(balnsmaxz,baldetmaxz2);
                baldeteffzmin=max(balnsminz,baldetminz2);

                -
                ((pz*(h/asd2))/tan(ThetaEmt*pi/180))+((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2)
                -((pz*(h/asd2))/tan(ThetaEmt*pi/180))-
                ((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2)

            ballistic_ns_int_temp=(1/(asd*asd2))*GeomFractionIntgtEmitIntgtDet(Asw/asd, -
            ((pz*(h/asd2))/tan(ThetaEmt*pi/180))+((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2), -
            ((pz*(h/asd2))/tan(ThetaEmt*pi/180))-((h/(2*asd2))/tan(ThetaEmt*pi/180))-sw/2),
            baldeteff_x, baldeteff

```

```

, (nsdetyhigh)+(baldeteffzmax/tan(nsThdet*pi/180)), (nsdetyhigh)+(baldeteffzmin/tan(nsTh
det*pi/180)), emtonsL+nslength, h, ThetaEmit, nsThdet, [1;1], [20;10],1);
    pause

    end
    ballistic_ns_int=ballistic_ns_int_temp+ballistic_ns_int;

end

end

end

%Transmission beneath the nanosheets
if (k==0)

    for pz=0.5:1:(asd2-0.5)

        pz+0.5;
        baldetminz = max(((sw-Bsh)*tan(ThetaDet*pi/180)), (nsdepth-
        ((pz*(h/asd2))-nsdepth)*((nstodetL+((sw-Bsh)/2))/(emtonsL+nslength-
        (sw/2)+((pz*(h/asd2))/tan(ThetaEmit*pi/180))))));
        baldetmaxz= min(h, (h-(((pz*(h/asd2))-h)*((nstodetL+((sw-
        Bsh)/2))/(emtonsL+nslength-(sw/2)+((pz*(h/asd2))/tan(ThetaEmit*pi/180))))));

        baldetminz2 = max(((sw-Bsh)*tan(ThetaDet*pi/180)), (nsdepth-
        ((pz*(h/asd2))-nsdepth)*((nstodetL+nslength+((sw-Bsh)/2))/(emtonsL-
        (sw/2)+((pz*(h/asd2))/tan(ThetaEmit*pi/180))))));
        baldetmaxz2= min(h, (h-(((pz*(h/asd2))-h)*((nstodetL+nslength+((sw-
        Bsh)/2))/(emtonsL-(sw/2)+((pz*(h/asd2))/tan(ThetaEmit*pi/180))))));

        %pause

        %pause

        if (baldetmaxz2 <= baldetminz)|| (baldetminz2>=baldetmaxz) %2nd
statement may not be possible but I will keep it there just in case.

            ballistic_of_int_temp=0;
            %fprintf(1,'didn't integrate due to z restriction for
bottom\n');

        else

            baldeteffzmax=min(baldetmaxz,baldetmaxz2);
            baldeteffzmin=max(baldetminz,baldetminz2);

ballistic_of_int_temp=(1/(asd2))*GeomFractionIntgtEmitIntgtDet(Asw, -
((pz*(h/asd2))/tan(ThetaEmit*pi/180))+((h/(2*asd2))/tan(ThetaEmit*pi/180))-sw/2), -
(((pz*(h/asd2))/tan(ThetaEmit*pi/180))-((h/(2*asd2))/tan(ThetaEmit*pi/180))-sw/2), 0,
Bsw, -((sw/2)-(baldeteffzmax/tan(ThetaDet*pi/180))),-((sw/2)-
(baldeteffzmin/tan(ThetaDet*pi/180))), mesalength+sw, h, ThetaEmit, ThetaDet, [asd;1],
[20;10],0);

        %pause

    end
    ballistic_of_int=ballistic_of_int_temp+ballistic_of_int;

end

end

end

```

```

    intforcas=emtons_int-ballistic_ns_int;
    casfac=1;
    intaftcas=intforcas*casfac;

    nstodet_int=GeomFractionIntgtEmitIntgtDet(nswidth, nsemtylow, nsemt high, -
(nspitch/2)-k*(nspitch), Bsw, Detylow, Detyhigh, nstodetL+((sw-Bsh)/2), h, nsThdet,
ThetaDet, [10;10], [20;10],1);

    k=k+1

    cumu_int=intaftcas*nstodet_int;
    cumu_int_bal=cumu_int+ballistic_int;
    int_bal_of=ballistic_int+ballistic_of_int;
    cumu_int_bal_of=cumu_int+int_bal_of;

    transmat(k,1)=nspitch/2+((k-1)*(nspitch));
    transmat(k,2)=emtons_int;
    transmat(k,3)=ballistic_ns_int;
    transmat(k,4)=intforcas;
    transmat(k,5)=casfac; % casimir ziman transmission factor can be set to 1 and
later calculated using
    transmat(k,6)=nstodet_int;
    transmat(k,7)=cumu_int;
    transmat(k,8)=ballistic_int;
    transmat(k,9)=ballistic_of_int;
    transmat(k,10)=cumu_int_bal;
    transmat(k,11)=Afract*cumu_int; %Casimir treated nanosheet transmission (phonons
hitting the wall of nanosheet)
    transmat(k,12)=Afract*ballistic_int; %Ballistic transmission through the
nanosheets only
    transmat(k,13)=Afract*ballistic_of_int; %Ballistic transmission under the
nanosheets only
    transmat(k,14)=Afract*cumu_int_bal; %Casimir treated nanosheet
transmission+Ballistic transmission through the nanosheets only
    transmat(k,15)=sum(transmat(:,11))*2;%Casimir treated nanosheet transmission
summed for all nanosheets and multiplied by 2.
    transmat(k,16)=(sum(transmat(:,12))*2)+transmat(1,13);%total ballistic
transmission
    transmat(k,17)=(sum(transmat(:,14))*2)+transmat(1,13);%(Casimir treated nanosheet
transmission+Ballistic transmission through the nanosheets only) summed for all
nanosheets and multiplied by 2 + Ballistic transmission under the nanosheets only

    emtons_int=0;
    ballistic_ns_int=0;
    intforcas=0;
    casfac=0;
    nstodet_int=0;
    cumu_int=0;
    ballistic_int=0;
    cumu_int_bal=0;

end

str=strcat(chipname, '_tmat_trial');
varname=genvarname(str);
assignin('base',varname, transmat);

geomfractnsmod6=[transmat(k,13),transmat(k,15),transmat(k,16)];

end

```

## REFERENCES

- <sup>1</sup> A. K. McCurdy, H. J. Maris, and C. Elbaum, Phys. Rev. B **2** (10), 4077 (1970).
- <sup>2</sup> H. J. Trumpp and W. Eisenmenger, Z. Phys. B **28** (3), 159 (1977).
- <sup>3</sup> O. Koblinger, J. Mebert, E. Dittrich, S. Dottinger, W. Eisenmenger, P. V. Santos, and L. Ley, Phys. Rev. B **35** (17), 9372 (1987).
- <sup>4</sup> J. B. Hertzberg, O. O. Otelaja, N. J. Yoshida, and R. D. Robinson, Rev. Sci. Instrum. **82** (10), 104905 (2011).
- <sup>5</sup> B. Taylor, H. J. Maris, and C. Elbaum, Phys. Rev. Lett. **23** (8), 416 (1969).
- <sup>6</sup> H. J. Maris, J. Acoust. Soc. Am. **50** (3), 812 (1971).
- <sup>7</sup> J. P. Wolfe, *Imaging phonons : acoustic wave propagation in solids*. (Cambridge University Press, Cambridge, U.K.; New York, 1998).
- <sup>8</sup> K. Jakata, Master's Thesis, University of the Witwatersrand, 2007.
- <sup>9</sup> K. Jakata and A. G. Every, Phys. Rev. B **77** (17), 174301 (2008).
- <sup>10</sup> S. Tamura, J. A. Shields, and J. P. Wolfe, Phys. Rev. B **44** (7), 3001 (1991).